ASC X9 TR 50–2019

Quantum Techniques in Cryptographic Message Syntax (CMS)

A Technical Report prepared by:
Accredited Standards Committee X9, Incorporated
Financial Industry Standards

Registered with American National Standards Institute

Date Registered: January 20, 2019

Contents

Page

Figures

Tables

Foreword

Publication of this Technical Report that has been registered with ANSI has been approved by the Accredited Standards Committee X9, Incorporated, 275 West Street, Suite 107, Annapolis, MD 21401. This document is registered as a Technical Report according to the "Procedures for the Registration of Technical Reports with ANSI." This document is not an American National Standard and the material contained herein is not normative in nature. Comments on the content of this document should be sent to: Attn: Executive Director, Accredited Standards Committee X9, Inc., 275 West Street, Suite 107, Annapolis, MD 21401,

Published by

Accredited Standards Committee X9, Incorporated
Financial Industry Standards
275 West Street, Suite 107
Annapolis, MD 21401 USA
X9 Online http://www.x9.org

Introduction

This Technical Report is a product of the Accredited Standards Committee X9 Financial Industry Standards, and was generated by the X9F4 workgroup.

The financial services industry is among the largest, most complex, and most consequential of any industry in the world. Nearly all people living in developed countries routinely utilize financial services. Be it for mortgages, loans, investments, money transfers, day-to-day banking, or any other of the myriad services available, the financial services industry greatly impacts the lives of those who use it. Beyond personal finances, these services are absolutely required by businesses, enterprise, government, and so on. The key point is that financial services are ubiquitous and indispensable. Those who use financial services generally do so under the assumptions that their information will be kept confidential, be secured against malicious use, and that only they or legitimately authorized entities have access to it.

Given the scale and complexity of their operations, for an enterprise operating within the financial services industry the task of ensuring security is not simple, not obvious, and not easy. In particular, developing, implementing, and distributing protection against new attacks (especially fundamentally new attacks) can be extremely complicated. Hence, it is in the best interest for those working within this sphere to be as proactive as possible in threat defence. If a threat can be predicted in advance of its realization, then time can be used to analyze the threat and prepare defences against it. Such a window of time would be particularly useful in the case of a major threat – which appears to be the case with quantum computers.

The invention of a large-scale quantum computer represents perhaps the biggest threat to cybersecurity in its history. Because the domain of potential targets for a quantum-capable attacker is so vast, the measures that will need to be taken to defend against their attacks will be great and will be varied.

This report serves to give the reader a general introduction to quantum computers and the consequences they pose to the financial services industry. Another purpose of this document is to give X9 members an understanding of the threats quantum computers pose to cybersecurity, and what some of the options are to mitigate those threats. Additionally, this report investigates the use of the Cryptographic Message Syntax (CMS) in the presence of a quantum-capable attacker and makes suggestions for using quantum-safe cryptography within the CMS, and for migrating classical systems to use quantum-safe algorithms.

Suggestions for the improvement or revision of this Technical Report are welcome. They should be sent to the X9 Committee Secretariat, Accredited Standards Committee X9, Inc., Financial Industry Standards, 275 West Street, Suite 107, Annapolis, MD 21401 USA.

This Technical Report was processed and registered for submittal to ANSI by the Accredited Standards Committee on Financial Services, X9. Committee approval of the Technical Report does not necessarily imply that all the committee members voted for its approval.

At the time this Technical Report was published, the X9 committee had the following members:

Roy C. DeCicco, X9 Chair
Angela Hendershott, X9 Vice Chair
Steve Stevens, Executive Director
Janet Busch, Program Manager

Organization Represented	*Representative*
ACI Worldwide	Doug Grote
Amazon	John Britton
American Bankers Association	Diane Poole
American Express Company	David Moore
Bank of America	Daniel Welch

BDO ....Jeffrey Ward
Bloomberg LP ....Corby Dear
Capital One ....Marie LaQuerre
Citigroup, Inc. ....Karla McKenna
Conexxus, Inc. ....Gray Taylor
CUSIP Global Services ....Gerard Faulkner
Delap LLP ....Andrea Beatty
Deluxe Corporation ....Angela Hendershott
Diebold Nixdorf ....Bruce Chapa
Dover Fueling Solutions ....Henry Fieglein
eCurrency ....David Wen
Federal Reserve Bank ....Mary Hughes
First Data Corporation ....Lisa Curry
FIS ....Stephen Gibson-Saxty
Fiserv ....Dan Otten
FIX Protocol Ltd - FPL ....James Northey
Futurex ....Ryan Smith
Gilbarco ....Bruce Welch
Harland Clarke ....John McCleary
Hyosung TNS Inc. ....Joe Militello
IBM Corporation ....Todd Arnold
ISARA Corporation ....Alexander Truskovsky
ISITC ....Lisa Iagatta
ITS, Inc. (SHAZAM Networks) ....Manish Nathwani
J.P. Morgan Chase ....Roy DeCicco
MagTek, Inc. ....Mimi Hart
MasterCard Europe Sprl ....Mark Kamers
NACHA The Electronic Payments Association ....George Throckmorton
National Security Agency ....Mike Boyle
NCR Corporation ....Kevin Spengler
Office of Financial Research, U.S. Treasury Department ....Thomas Brown Jr.
PCI Security Standards Council ....Troy Leach
RouteOne ....Chris Irving
SWIFT/Pan Americas ....Karin DeRidder
Symcor Inc. ....Debbi Fitzpatrick
TECSEC Incorporated ....Ed Scheidt
The Clearing House ....Sharon Jablon
U.S. Bank ....Michelle Wright
U.S. Commodity Futures Trading Commission (CFTC) ....Robert Stowsky
USDA Food and Nutrition Service ....Kathy Ottobre
VeriFone, Inc. ....Dave Faoro
Viewpointe ....Richard Luchak
VISA ....Kim Wagner
Wells Fargo Bank ....Mark Schaffer

At the time this Technical Report was published, the X9F Data and Information subcommittee on had the following members:

Dave Faoro, Chairman
Steven Bowles, Vice-Chairman

Organization Represented	***Representative***
ACI Worldwide	Doug Grote
ACI Worldwide	Dan Kinney

ACI Worldwide .... Lisa McKee
ACI Worldwide .... Julie Samson
American Bankers Association .... Tom Judd
American Express Company .... Gail Chapman
American Express Company .... Farid Hatefi
American Express Company .... David Moore
American Express Company .... John Timar
American Express Company .... Kevin Welsh
Bank of America .... Amanda Adams
Bank of America .... Peter Capraro
Bank of America .... Andi Coleman
Bank of America .... Joel Kazin
Bank of America .... Terry McKinney
Bank of America .... Matthew Sharp
Bank of America .... Michael Smith
Bank of America .... Daniel Welch
BDO .... Tim Crawford
BDO .... Jeffrey Ward
BlackBerry Limited .... Daniel Brown
Bloomberg LP .... Erik Anderson
Bloomberg LP .... Corby Dear
Capital One .... Marie LaQuerre
Capital One .... Johnny Lee
Cipherithm .... Scott Spiker
comforte AG .... Thomas Gloerfeld
comforte AG .... Henning Horst
Communications Security Establishment .... Jonathan Hammell
Communications Security Establishment .... David Smith
Conexxus, Inc. .... David Ezell
Conexxus, Inc. .... Alan Thiemann
CUSIP Global Services .... Scott Preiss
Deixis, PBC .... Pamela Bell
Deixis, PBC .... Allen Brown
Deixis, PBC .... Pat Santos
Delap LLP .... Andrea Beatty
Delap LLP .... David Buchanan
Deluxe Corporation .... Angela Hendershott
Deluxe Corporation .... Margiore Romay
Deluxe Corporation .... Andy Vo
Diebold Nixdorf .... Christoph Bruecher
Diebold Nixdorf .... Andrea Carozzi
Diebold Nixdorf .... Bruce Chapa
Diebold Nixdorf .... Michael Nolte
Diebold Nixdorf .... Michael Ott
Diebold Nixdorf .... David Phister
Digicert .... Tim Hollebeek
Digicert .... Steve Medin
Discover Financial Services .... Cheryl Mish
Discover Financial Services .... Diana Pauliks
Discover Financial Services .... Jordan Schaefer
Discover Financial Services .... Jorge Vargas
Dover Fueling Solutions .... Henry Fieglein
eCurrency .... David Wen
Federal Reserve Bank .... Patrick Adler
Federal Reserve Bank .... Guy Berg
Federal Reserve Bank .... Marianne Crowe

Federal Reserve Bank .... Amanda Dorphy
Federal Reserve Bank .... Mary Hughes
Federal Reserve Bank .... Daniel Maynard
Federal Reserve Bank .... Susan Pandy
Federal Reserve Bank .... Patti Ritter
First Data Corporation .... Lisa Curry
First Data Corporation .... Kalli Davidson
First National Bank of Omaha .... Sherry Rewolinski
First National Bank of Omaha .... Kristi White
FIS .... Saman Amighi
FIS .... John Soares
FIS .... Sunny Wear
Fiserv .... Bud Beattie
Fiserv .... Dan Otten
Futurex .... Steven Bowles
Futurex .... Ryan Smith
Futurex .... Tim Weston
GEOBRIDGE Corporation .... Donna Gem
GEOBRIDGE Corporation .... Jason Way
Gilbarco .... Scott Turner
Gilbarco .... Bruce Welch
Harland Clarke .... Joseph Filer
Heartland Payment Systems .... Scott Meeker
Heartland Payment Systems .... Govindaraj Palanisamy
Hyosung TNS Inc. .... Joe Militello
Hyosung TNS Inc. .... Jay Shin
IBM Corporation .... Todd Arnold
IBM Corporation .... Richard Kisley
Ingenico .... Wayne Burgess
Ingenico .... Nabil Hamzi
Ingenico .... Steve McKibben
ISARA Corporation .... Mike Brown
ISARA Corporation .... Philip Lafrance
ISARA Corporation .... Alexander Truskovsky
ITS, Inc. (SHAZAM Networks) .... Janet LaFrence
ITS, Inc. (SHAZAM Networks) .... Manish Nathwani
J.P. Morgan Chase .... Kathleen Krupa
J.P. Morgan Chase .... Jackie Pagán
J.P. Morgan Chase .... Darryl Scott
Level 10 .... Allan Elder
MagTek, Inc. .... Jeff Duncan
MagTek, Inc. .... Mimi Hart
MasterCard Europe Sprl .... Mark Kamers
MasterCard Europe Sprl .... Larry Newell
MasterCard Europe Sprl .... Adam Sommer
MasterCard Europe Sprl .... Michael Ward
Member Emeritus .... Lawrence LaBella
Member Emeritus .... Mike McCormick
Micro Focus .... Luther Martin
National Institute of Standards and Technology (NIST) .... Elaine Barker
National Institute of Standards and Technology (NIST) .... Lily Chen
National Security Agency .... Mike Boyle
National Security Agency .... Paul Timmel
NCR Corporation .... Charlie Harrow
NCR Corporation .... Bradford Loewy
P97 Networks, Inc. .... Steve Moses

PCI Security Standards Council .... Leon Fell
PCI Security Standards Council .... Troy Leach
PCI Security Standards Council .... Ralph Poore
RSA, The Security Division of EMC .... Steve Schmalz
SafeNet Infotech Pvt. Ltd. .... Amit Sinha
SafeNet Infotech Pvt. Ltd. .... Devesh Tewari
TECSEC Incorporated .... Ed Scheidt
TECSEC Incorporated .... Dr. Wai Tsang
TECSEC Incorporated .... Jay Wack
Thales UK Limited .... Larry Hines
Thales UK Limited .... James Torjussen
The Clearing House .... Mark Fitlin
The Clearing House .... Sharon Jablon
The Clearing House .... Hirak Patel
The Clearing House .... Miguel Sanchez
The Phoenix Group .... Ron Davis
The Phoenix Group .... Candice Hoft
Trustwave .... John Amaral
U.S. Bank .... Stephen Case
University Bank .... Stephen Ranzini
University Bank .... Michael Talley
VeriFone, Inc. .... John Barrowman
VeriFone, Inc. .... Dave Faoro
VeriFone, Inc. .... Doug Manchester
VeriFone, Inc. .... Brad McGuinness
VeriFone, Inc. .... Saxon Noh
VeriFone, Inc. .... Joachim Vance
VISA .... Ben Choong
VISA .... Eric Le Saint
VISA .... Kim Wagner
Wells Fargo Bank .... Allen Ausec
Wells Fargo Bank .... David Cooper
Wells Fargo Bank .... William Felts, IV
Wells Fargo Bank .... Matthew Greenwell
Wells Fargo Bank .... Phillip Griffin
Wells Fargo Bank .... Calvin Heng
Wells Fargo Bank .... Jan Kohl
Wells Fargo Bank .... Garrett Macey
Wells Fargo Bank .... Kelly O'Donnell
Wells Fargo Bank .... Olatunde Ojolola
Wells Fargo Bank .... Mark Schaffer
Wells Fargo Bank .... Maria Schuett
Wells Fargo Bank .... Jeff Stapleton
Wells Fargo Bank .... Tony Suarez
White and Williams LLP .... Richard Borden
White and Williams LLP .... Sandra Lambert
White and Williams LLP .... Joshua Mooney
White and Williams LLP .... Michael Olsan

Under ASC X9, Inc. procedures, a working group may be established to address specific segments of work under the ASC X9 Committee or one of its subcommittees. A working group exists only to develop standard(s) or technical report(s) in a specific area and is then disbanded. The individual experts are listed with their affiliated organizations. However, this does not imply that the organization has approved the content of the standard or technical report.

At the time this Technical Report was published, the X9F4 Cybersecurity and Cryptographic Solutions workgroup which developed this technical report had the following active members:

Jeff Stapleton, Chairman
Sandra Lambert, Vice-Chairman
Philip Lafrance, Project Editor

Organization Represented	***Representative***
ACI Worldwide	Lisa McKee
Amazon	John Britton
Amazon	Igor Kleyman
Amazon	Rahul Prabhakar
American Express Company	Gail Chapman
American Express Company	David Moore
Bank of America	Amanda Adams
Bank of America	Peter Capraro
Bank of America	Andi Coleman
Bank of America	Joel Kazin
Bank of America	Terry McKinney
Bank of America	David Mortman
Bank of America	Matthew Sharp
Bank of America	Daniel Welch
BlackBerry Limited	Daniel Brown
Bloomberg LP	Erik Anderson
Capital One	Johnny Lee
Cipherithm	Scott Spiker
comforte AG	Henning Horst
Conexxus, Inc.	David Ezell
Conexxus, Inc.	Alan Thiemann
Conexxus, Inc.	Linda Toth
Deixis, PBC	Pamela Bell
Deixis, PBC	Allen Brown
Deixis, PBC	Pat Santos
Delap LLP	Andrea Beatty
Delap LLP	Spencer Giles
Diebold Nixdorf	Christoph Bruecher
Diebold Nixdorf	Rick Brunt
Diebold Nixdorf	Andrea Carozzi
Diebold Nixdorf	Bruce Chapa
Diebold Nixdorf	Scott Harroff
Diebold Nixdorf	Anne Konecny
Diebold Nixdorf	Michael Nolte
Diebold Nixdorf	Michael Ott
Diebold Nixdorf	David Phister
Diebold Nixdorf	Matthias Runowski
Digicert	Tim Hollebeek
Digicert	Steve Medin
Discover Financial Services	Cheryl Mish
Discover Financial Services	Diana Pauliks
Discover Financial Services	Lakshmi Ramanathan
Discover Financial Services	Jordan Schaefer
Discover Financial Services	Jorge Vargas
Dover Fueling Solutions	Henry Fieglein
Federal Reserve Bank	Patrick Adler
Federal Reserve Bank	Guy Berg

Federal Reserve Bank....................Marianne Crowe
Federal Reserve Bank....................Pieralberto Deganello
Federal Reserve Bank....................Amanda Dorphy
Federal Reserve Bank....................Mary Hughes
Federal Reserve Bank....................Daniel Littman
Federal Reserve Bank....................Daniel Maynard
Federal Reserve Bank....................Paul Nunnally
Federal Reserve Bank....................Susan Pandy
Federal Reserve Bank....................Patti Ritter
First Data Corporation....................Lisa Curry
First Data Corporation....................Kalli Davidson
First Data Corporation....................Prince Duodu
First Data Corporation....................Brian Kean
First Data Corporation....................Brian Murray
First Data Corporation....................Randall Rieth
First Data Corporation....................Allisa Walker
FIS....................Saman Amighi
FIS....................Ian Lumsden
FIS....................Sunny Wear
Fiserv....................Dan Otten
FIX Protocol Ltd - FPL....................James Northey
Futurex....................Steven Bowles
GEOBRIDGE Corporation....................Donna Gem
GEOBRIDGE Corporation....................Dean Macinskas
GEOBRIDGE Corporation....................Jason Way
Gilbarco....................Bruce Welch
Harland Clarke....................John McCleary
Heartland Payment Systems....................Dipti Desai
Heartland Payment Systems....................Govindaraj Palanisamy
IBM Corporation....................Todd Arnold
IBM Corporation....................Richard Kisley
ISARA Corporation....................Mike Brown
ISARA Corporation....................Philip Lafrance
ISARA Corporation....................Alexander Truskovsky
ITS, Inc. (SHAZAM Networks)....................Janet LaFrence
ITS, Inc. (SHAZAM Networks)....................Manish Nathwani
J.P. Morgan Chase....................Roger Cowie
J.P. Morgan Chase....................Robin Doyle
J.P. Morgan Chase....................Clinton Jones
J.P. Morgan Chase....................Jackie Pagán
J.P. Morgan Chase....................Darryl Scott
K3DES LLC....................Davi Ottenheimer
Level 10....................Allan Elder
MagTek, Inc.....................Mimi Hart
MasterCard Europe Sprl....................Leland Englebardt
MasterCard Europe Sprl....................Mark Kamers
Member Emeritus....................Darlene Kargel
Member Emeritus....................Gene Kathol
Member Emeritus....................Lawrence LaBella
Member Emeritus....................Mike McCormick
Member Emeritus....................Bill Poletti
Member Emeritus....................Richard Sweeney
Member Emeritus....................Mark Tiggas
National Institute of Standards and Technology (NIST)....................Elaine Barker
National Institute of Standards and Technology (NIST)....................Lily Chen
National Institute of Standards and Technology (NIST)....................Burak Sahin

National Security Agency ....Greg Gilbert
National Security Agency ....Tim Havighurst
National Security Agency ....Paul Timmel
NCR Corporation ....Charlie Harrow
NCR Corporation ....Bradford Loewy
NCR Corporation ....Brian Wotherspoon
PCI Security Standards Council ....Leon Fell
PCI Security Standards Council ....Troy Leach
PCI Security Standards Council ....John Markh
PCI Security Standards Council ....Ralph Poore
PCI Security Standards Council ....Elizabeth Terry
RSA, The Security Division of EMC ....Steve Schmalz
SafeNet Infotech Pvt. Ltd..Amit Sinha
TECSEC Incorporated..Ed Scheidt
TECSEC Incorporated..Dr. Wai Tsang
TECSEC Incorporated..Jay Wack
Thales UK Limited ....Colette Broadway
Thales UK Limited ....Larry Hines
Thales UK Limited ....James Torjussen
The Clearing House..Ken Friedman
The Clearing House..Sharon Jablon
The Clearing House..Miguel Sanchez
U.S. Bank..Stephen Case
U.S. Bank..Steven Fenter
U.S. Bank..Darin Pettis
University Bank...Stephen Ranzini
University Bank...Michael Talley
VeriFone, Inc. ....Dave Faoro
VeriFone, Inc. ....LeAnn Hostetler
VeriFone, Inc. ....Doug Manchester
VeriFone, Inc. ....Saxon Noh
VeriFone, Inc. ....Joachim Vance
VISA..Ben Choong
VISA..Adam Clark
VISA..Marcelo Silva
VISA..Johan ("Hans") Van Tilburg
VISA..Kim Wagner
Wells Fargo Bank ....Sotos Barkas
Wells Fargo Bank ....William Felts, IV
Wells Fargo Bank ....Matthew Greenwell
Wells Fargo Bank ....Phillip Griffin
Wells Fargo Bank ....Sam Grosby
Wells Fargo Bank ....Jeff Jacoby
Wells Fargo Bank ....Joseph Kaluzny
Wells Fargo Bank ....Brian Keltner
Wells Fargo Bank ....Jan Kohl
Wells Fargo Bank ....Eric Lengvenis
Wells Fargo Bank ....Olatunde Ojolola
Wells Fargo Bank ....Doug Pelton
Wells Fargo Bank ....Maria Schuett
Wells Fargo Bank ....Jeff Stapleton
Wells Fargo Bank ....Tony Stieber
Wells Fargo Bank ....Tony Suarez
Wells Fargo Bank ....Nathan Suri
White and Williams LLP..Gwenn Barney
White and Williams LLP..Richard Borden

White and Williams LLP..Clay Epstein
White and Williams LLP..Michael Jervis
White and Williams LLP..Sandra Lambert
White and Williams LLP..Andrew Lipton
White and Williams LLP..Joshua Mooney
White and Williams LLP..Michael Olsan

Quantum Techniques in Cryptographic Message Syntax (CMS)

1 Scope

This technical report provides information about quantum computers and post-quantum cryptography for people working in the financial services industry. In particular, this report investigates how a large-scale quantum computer could impact the security of commonly used protocols within the CMS and makes recommendations to mitigate those impacts.

This report achieves its goal by first discussing the basics of quantum computers and the quantum algorithms that break classical cryptography, and then shows how those algorithms could be used to attack classical cryptosystems. Moreover, this report provides a basic background in each of the main branches of mathematics which are thought to yield quantum-safe cryptographic schemes.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

2.1 X9.73:2017 Cryptographic Message Syntax (CMS) – ASN.1 and XML

2.2 X9.98:2010 Lattice-Based Polynomial Public Key Establishment Algorithm

2.3 ISO 16609:2012 Banking – Requirements for message authentication using symmetric techniques

2.4 FIPS 180-4 Secure Hash Standard (SHS)

2.5 FIPS 197 Advanced Encryption Standard (AES)

2.6 FIPS 198-1 The Keyed-Hash Message Authentication Code (HMAC)

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1
Advanced Encryption Standard (AES)

A symmetric encryption algorithm defined by FIPS PUB 197. With an appropriate mode of operation, it can provide privacy (encryption) and integrity validation. It is believed that, assuming a random 256 bit key, AES encrypted texts are secure against Quantum Computers.

3.2
Asymmetric cryptographic algorithm

A cryptographic algorithm that has two related keys, a public key and a private key; the two keys have the property that, given the public key, it is computationally infeasible to derive the private key.

3.3
Certificate
Digital certificate

The public key and identity of an entity, together with some other information, that is rendered unforgeable by signing the certificate with the private key of the Certification Authority that issued the certificate.

3.4
Certificate Authority
CA

The entity trusted by one or more other entities to create and assign certificates.

3.5
Content Encryption Key
CEK

Symmetric key used to encrypt the content of a message.

3.6
Cryptanalysis

The study of analyzing information systems in order to study the hidden aspects of the systems.

3.7
Cryptographic hash function
Hash function

A (mathematical) function that maps values from a large (possibly very large) domain into a smaller range and satisfies the following properties:

(One-way) It is computationally infeasible to find any input that map to any pre-specified output;

(Collision Resistance) It is computationally infeasible to find any two distinct inputs that map to the same output.

3.8
Cryptographic key
Key

A parameter that determines, possibly with other parameters, the operation of a cryptographic function such as:

the transformation from plaintext to ciphertext and vice versa;

the synchronized generation of keying material;

digital signature computation or validation.

3.9
Cryptography

The discipline that embodies principles, means and methods for the transformation of data to hide its information content, prevent its undetected modification, and prevent its unauthorized use or a combination thereof.

3.10
Digital Signature

An electronic signature based on cryptographic rules and parameters of originator authentication, which identify the signer and verify the integrity of the data pertaining to the signature.

3.11
Key agreement

Method of establishing a key, whereby both parties contribute to the value of the resulting key and neither party can control the value of the resulting key.

3.12
Key Derivation Function
KDF

A mathematical function which derives one or more secret keys from a secret value such as a master key, a password, or a passphrase using a pseudorandom function.

3.13
Key Encryption Key
KEK

Key used exclusively to encrypt and decrypt keys.

3.14
Keying material

Data (e.g., keys, certificates and initialization vectors) necessary to establish and maintain cryptographic keying relationships.

3.15
Key management

Generation, storage, secure distribution and application of keying material in accordance with a security policy.

3.16
Key pair

A public key and its corresponding private key used in public key cryptography.

3.17
Key transport

Key establishment protocol under which the secret key is determined by the initiating party.

3.18
Message Authentication Code
MAC

Cryptographic value that is the result of passing a message through the message authentication algorithm using a specific key.

3.19
Multipurpose Internet Mail Extensions
MIME

Format for internet message bodies as defined in the IETF documents RFCs (2045-7, 2049, 2184, 2231, 3023, and 4288-9).

3.20
Private key

In an asymmetric (public) key cryptosystem, the key of an entity's key pair that is known only by that entity

NOTE A private key may be used to compute the corresponding public key, to make a digital signature that may be verified by the corresponding public key, to decrypt data encrypted by the corresponding public key; or together with other information to compute a piece of common shared secret information.

3.21
Public key

That key of an entity's key pair that may be publicly known in an asymmetric (public) key cryptosystem.

NOTE A public key may be used to verify a digital signature that is signed by the corresponding private key, to encrypt data that may be decrypted by the corresponding private key, or by other parties to compute shared information

3.22
Random Access Memory
RAM

A form of computer data storage that stores data and machine code currently in use.

3.23
Secure MIME
S/MIME

Specification for handling MIME data securely by adding cryptographic security services to supply authentication, message integrity, non-repudiation of origin, privacy and data security

3.24
Shared symmetric key

Symmetric key derived from a shared secret value and other information

3.25
Static key

Private or public key that is common to many executions of a cryptographic scheme.

3.26
Symmetric cryptographic algorithm

Cryptographic algorithm that uses one shared, secret, key.

NOTE The key shall be kept secret between the two communicating parties, and the same symmetric key that is used for encryption is used for decryption.

3.27
Symmetric key

Cryptographic key that is used in symmetric cryptographic algorithms.

NOTE The same symmetric key that is used for encryption is also used for decryption.

3.28
Feistel cipher (a.k.a. Feistel network)

A symmetric structure used in the construction of block ciphers, named after the German IBM cryptographer Horst Feistel and commonly known as the Feistel network. Many block ciphers use the scheme, including the Data Encryption Standard.

4 Symbols and abbreviated terms

For the purposes of this document, the following symbols and abbreviations apply.

AES Advanced Encryption Standard

CEK Content Encryption Key

CMS Cryptographic Message Syntax

CVP Closest Vector Problem

DES Data Encryption Standard

DLOG Discrete Logarithm Problem

ECC Elliptic Curve Cryptography

ECDH Elliptic Curve Diffie-Hellman

FTP File Transfer Protocol

FTS Few Time Signature

HFE Hidden Field Equation

HMAC Keyed-Hash Message Authentication Code

HSP Hidden Subgroup Problem

HTTPS Secure Hypertext Transfer Protocol

IPsec Internet Protocol Security

KEK Key Encryption Key

KEM Key Encapsulation Mechanism

L2TP Layer 2 Tunnelling Protocol

LWE	Learning With Errors
MAC	Message Authentication Code
NFS	Number Field Sieve
OTS	One Time Signature
PKI	Public Key Infrastructure
QKD	Quantum Key Distribution
RAM	Random Access Memory
RSA	Rivest Shamir Adelman encryption scheme
S/MIME	Secure Multipurpose Internet Mail Extensions
SHA2	Secure Hash Algorithm 2
SIDH	Supersingular Isogeny Diffie-Hellman
SIS	Short Integer Solution
SMTP	Simple Mail Transfer Protocol
TLS	Transport Layer Security
VPN	Virtual Private Network

5 Quantum Computers

This section provides a general introduction to quantum computing as it relates to industry and cybersecurity. This section does not attempt to give a comprehensive overview of quantum mechanics, quantum computation, or the state-of-the-art in any related field. Instead, this section discusses some of the more impactful non-cryptographic potential uses for quantum computers and then describes the most damaging effects quantum computers will have on enterprise security and cybersecurity in general. Below is a high-level description of what quantum computing is and how quantum computers differ from classical computers.

Classical computers are devices that encode information onto some sort of physical system, and then performs operations on that encoded information according to some set of rules. For example, one may store information by polarizing ferromagnetic materials, (as in a hard disk drive) or by charging or discharging a capacitor (as in RAM). One may then transform this data according to a specified set of rules, called a computer program, or algorithm.

Quantum Computers extend this notion of computation by adding quantum bits (qubits) to the system. An algorithm can then perform operations on the quantum mechanical components of the computer as well as the classical components.

The introduction of quantum mechanical components to an otherwise classical system is such a powerful idea because, quantum mechanics has several fundamental properties that, when carefully taken advantage of, allow for computations which would not be possible on a classical computer. These properties include: superposition, entanglement, and interference. In many cases, even a large cluster of classical computers working in parallel cannot compete with a single quantum computer. For a more technical introduction, see [1] and [2].

5.1 Non-cryptographic Use Cases for Quantum Computers

Quantum computers open the doors to potential new technologies that, if realized, will have massive positive benefits to industry and society. These technologies can be realized by either a universal quantum computer, or in some cases, by more specialized quantum-enabled devices. Below are some of the most impactful (non-cryptographic) uses for these devices.

5.1.1 Proprietary Drug Design

Even relatively simple problems in chemistry can be very difficult to solve on a classical computer. Moreover, the complexity of these problems increases dramatically even if only a few extra atoms are considered. For example, "exactly computing the energies of methane (CH_4) takes about one second, but the same calculation takes about ten minutes for ethane (C_2H_6) and about ten days for propane (C_3H_8)" [5]. Hence, simulating and modelling very complex chemical interactions is not a tractable problem on classic architectures. However, it is thought that these types of simulations could be carried out by a large-scale quantum computer. Drug design thus gains considerable improvement by harnessing the powers of quantum computation [3] and [4].

One possible use case for such complex modelling is using an individual's genomic data to design pharmaceuticals specialized for that person's body. It is thought that such proprietary drugs would be more effective than their generic counterparts.

5.1.2 Material Science

Conventionally, superconductive materials need to be kept extremely cold to reach a superconductive state. In fact, if such a state is reached when liquid nitrogen is used as the coolant, then that conductor is generally thought of as a high-temperature superconductor. Much colder temperatures are often required for a material to reach a superconductive state.

Superconductive materials experience substantially less energy loss than non-superconductive alternatives and so, if one uses room temperature superconductive materials they may, for example, realize less costs and increase efficiency in their enterprise. One particularly attractive use for such materials is in electrical power lines; It is estimated that in between 2011 and 2015, about 5% of energy was lost while being transmitted over the US electrical grid per year.

It is not yet clear if materials that are superconductive at room temperature can be constructed at scale; and indeed, the proposition that such materials could ever be constructed has generated some controversy. However, clear progress towards high-temperature superconductors has been made [7][8][9]. It is conceivable then that with a quantum computer, one could model the complex physics necessary to design, for example, transmission lines which can retain a larger percentage of energy.

5.1.3 Big Data and Unstructured Searches

Many modern organizations, enterprises, and governmental bodies rely on vast quantities of data for their day to day operations. The data in these reserves is not always specialized to particular areas but can instead be seemingly arbitrary and unstructured. Massive reserves of data are unwieldy and the problems of finding a specific datum within the set, or performing analyses of the data become very difficult. Quantum search algorithms such as the generalized version of Grover's Algorithm [22] (cf. subsection 5.2.2), or quantum-aided machine learning techniques [12] lend themselves nicely to these sorts of problems.

5.1.4 Machine Learning and Artificial Intelligence

Quantum algorithms apply nicely to the fields of Machine Learning and Artificial Intelligence. For example, a quantum computer would accelerate the rate at which a neural network learns, resulting in more intelligent systems better capable at for example, risk assessment in the finance industry [11].

5.2 Impact to Cryptography

Quantum computers can enable attacks on cryptosystems that are not feasible with classical computers. It is important to understand the quantum algorithms that threaten classic cryptosystems, and the improvements that can be made to prevent those attacks.

5.2.1 Shor's Algorithm

Consider the following two problems:

1) Given an odd, composite integer N that is not a prime power[1], find a prime factor of N.
2) Given a generator g of a finite group G, and an element $g^k \in G$, find k.[2]

The first problem is known as the *Integer Factorization problem,* and the second is the *Discrete Logarithm (DLOG) problem*. Both problems are easy to understand (assuming some knowledge of group theory for the latter) but are difficult to solve in practice in general. Of course, not every integer that meets the three conditions above (1) is difficult to factor, and not every group (2) is difficult to calculate discrete logs in; stricter conditions must be imposed on these objects for use in cryptographic settings. Large integers and groups for which these problems are thought to be intractable form the basis for secure instances of most classic public-key cryptosystems.

It is unknown if there exist classical polynomial-time (efficient) algorithms for solving either of these problems. The fastest known classic algorithm for factoring integers is the Number Field Sieve (NFS) [18], which runs in sub-exponential time. The fastest known classic algorithm for computing discrete logs is Pollard's Rho algorithm [19] which runs in time $O(\sqrt{N})$.

The presumed intractability of these problems forms the security basis for the most widely deployed cryptosystems in the world today such as: RSA signatures and encryption, ECDSA, and ECDH. Secure protocols such as TLS and IPsec rely on RSA or ECDSA signatures to authenticate peers, and ECDH or RSA to establish shared keys between those peers.

Shor's Algorithm — named for its inventor Peter Shor — is a polynomial-time quantum algorithm for solving the Integer Factorization problem [20]. Moreover, a modified version of Shor's algorithm can be used to efficiently calculate discrete logs as well. Thus, when a sufficiently large universal quantum computer becomes available, nearly all currently deployed public-key cryptography becomes vulnerable to attack. Subsection 5.2.3 discusses how quantum algorithms will impact the security of symmetric-key cryptosystems and hash functions.

Shor's algorithm consists of two essential components: a classic reduction from the factorization problem to the order-finding problem[3], and a quantum algorithm for solving the order-finding problem. The complexity of Shor's algorithm has been found to be $O\left(\log(\mathrm{N})^2\left(\log\log(\mathrm{N})\right)\left(\log\log\log(\mathrm{N})\right)\right)$ [21].

---

[1] That is, $N \neq p^k$ for any prime p and positive integer k. Also, note that all three conditions are easy to verify.

[2] Implicitly k is a positive integer.

[3] Given an element g in a group G, find the least positive integer t such that g^t=1 in G.

5.2.2 Grover's Algorithm

Consider the following problem. Suppose there is a set of data where one piece of data in the set is special in some specified way; the problem is to find that datum. At a very high level, symmetric-key algorithms and hash functions are presumed intractable instances of this search problem. For example, given an instance of AES-256 (which uses 256-bit keys) and a known plaintext/ciphertext pair (m, c), find a key $k \in \{0,1\}^n$ such that $AES256^{-1}(k, c) = m$; find a key that decrypts c to m. In the case of hash functions, the problem becomes finding an input that maps to some specified output.

Grover's algorithm — named for its inventor Lov Kumar Grover — is a probabilistic quantum algorithm which solves this search problem [22]. More specifically, with high probability, Grover's search algorithm returns the special value in $O(2^{\frac{n}{2}})$ operations; a square root speed-up over classic methods. More generally, if there are p distinct values in the dataset that are special, then Grover's algorithm needs only $O(2^{\frac{n}{2}}p^{-\frac{1}{2}})$ operations to find one of them. Grover's algorithm is provably asymptotically optimal [22].

5.2.3 Symmetric and Asymmetric Key Lengths

In the past, whenever new attacks against RSA or ECC emerged, they were often mitigated by extending the lengths of the public/private keys. Likewise, as cryptanalysis improved against symmetric key schemes such as AES, the private keys needed to be adjusted to maintain security levels. Hence, the question arises: why can't we do the same thing when quantum computers arrive? The answer is that because Shor's algorithm is so efficient, keys long enough to be secure against it would be too large to be of practical use. And so, lengthening keys is not a viable solution.

The above applies to public-key cryptosystems; the case is a bit different for symmetric-key cryptosystems and hash functions. These schemes are in general not based on hard math problems (and hence have no associated security reductions), but rather on heuristically secure components and subroutines such as: substitution-permutation networks (Feistel networks), compression functions, bitmasks, bit operations, row/column operations, and so forth. Secure symmetric-key cryptosystems are computationally indistinguishable from uniformly random functions.

Because they are not based on hard problems, attacks on symmetric-key schemes historically involve attacking the components of the scheme. For example, by carefully examining potential biases in S-Box permutations (i.e., by investigating how internal pieces of the scheme deviate from being uniformly random) it is sometimes possible to find a relationship between plaintexts and their corresponding ciphertexts which probabilistically leads to a recovery of (pieces of) the private key. The two main types of such attacks are Linear and Differential Cryptanalysis. Analyses like these are very difficult to perform (although not impossible [23]) and require a large number of known plaintext/ciphertext pairs to succeed [24]. The symmetric-key schemes used in practice resist the best know cryptanalytic attacks.

However, symmetric-key schemes fit into the problem framework which Grover's algorithm attempts to solve; to search for a private key in the set of all possible keys. To see more concretely how Grover affects symmetric key lengths, consider an instance of AES-256. There are 2^{256} possible AES-256 keys. With high probability, Grover's algorithm will find the private key in approximately $= 2^{128}$ queries. In other words, Grover finds the private AES-256 key in an expected 2^{128} queries. Stated differently, AES-256 provides 128 bits of quantum security; half of the classical security level. Being a bit crude, the quantum security level offered by a symmetric-key cryptosystem (or a hash function) is about half of its classical security level. Hence, symmetric keys (or outputs of hash functions) need to be roughly doubled in length to maintain their current security levels.

It should be mentioned that finding preimages is not the only type of attack against hash functions. If it is intractable to find a preimage under a given hash function for a given (random) output element, then that hash function has the one-way property; often referred to as preimage-resistance. Other security properties for hash functions include collision and second-preimage resistance.

Different cryptographic schemes rely on different combinations of these security properties, such as one-wayness with collision resistance (or sometimes less studied properties such as subset-resilience). The details are omitted here, but in the classical setting, a hash function with n-bit output offers $n/2$-bits of security against collision attacks, and n-bits against second-preimage attacks.

The fasted known quantum attack on collision-resistance has time complexity $O(2^{n/3})$ [25], but as argued by Bernstein et. al. [27], when taking into account the storage requirements it is not better than the best known classical collision attack (van Oorschot-Weiner [28]) which has cost $O(2^{n/2})$. Grover's algorithm can also be used against second-preimage resistance; giving a quadratic speed-up over classical attacks. Many schemes rely only on the one-wayness of the underlying hash function.

Tables 1 and 2 below gives the classic and quantum security levels for some of the most widely used encryption schemes and hash functions. The bit strengths shown for SHA-256 and SHA-512 are against collision attacks. In Table 1 the notation "~0" is used to denote that the quantum security provided by that algorithm is practically zero.

Algorithm	Key Length	Classical Bit Strength	Quantum Bit Strength	Fastest Known Quantum Algorithm
RSA-2048	2048 bits	112 bits	~0 bits	Shor
RSA-3096	3096 bits	128 bits	~0 bits	Shor
ECC-256	256 bits	128 bits	~0 bits	Shor
ECC-512	512 bits	256 bits	~0 bits	Shor

Table 1: Key Lengths vs Security Levels Using Shor's Algorithm

Algorithm	Key Length	Classical Bit Strength	Quantum Bit Strength	Fastest Known Quantum Algorithm
AES-128	128 bits	128 bits	64 bits	Grover
AES-256	256 bits	256 bits	128 bits	Grover
SHA-256	256 bits	256 bits	128 bits	Grover
SHA-512	512 bits	512 bits	256 bits	Grover

Table 2: Key Lengths vs Security Levels Using Grover's Algorithm

5.3 Impact to Secure Network Communication

The security of widely deployed internet protocols such as TLS, CMS, and IPsec is severely compromised in the presence of a quantum-capable adversary. The security of these protocols is currently based on the classically hard number-theoretic problems discussed in Section 5.2. To provide security against a quantum-capable adversary these protocols will need to be secured with quantum-safe algorithms. This section describes in more detail the problems quantum computers pose to secure network communications in terms of interoperability, security, and upgrade approaches.

One of the most commonly used network communication protocols is Transport Layer Security. TLS is a client and server connection protocol consisting of two components: the handshake, and data exchanges. In the handshake portion of the protocol a cipher suite is negotiated, the client will authenticate the server, and the server might authenticate the client), and a shared secret is established. Both sides then derive common symmetric session keys for the data exchange. Often the client is another server establishing a TLS connection between two applications running on the servers.

The authentication portion of the handshake is most often done using RSA digital signatures, and the shared secret is generally established using RSA key transport, Diffie-Hellman (DH), or ephemeral Elliptic Curve Diffie-Hellman (ECDHE) algorithm. As factoring and discrete log calculations are done efficiently with Shor's algorithm, a quantum capable adversary may be able to forge RSA signatures or recover secret keying material.

A security compromise of TLS implies a security compromise of any data transfer protocol which uses TLS. Examples of such protocols include: HTTPS, FTP, SMTP, and L2TP. A security compromise of a protocol like HTTPS would leave most Internet users vulnerable. Protocols such as L2TP are used in VPNs, which are popular in enterprise PKI. Thus, employees working remotely may be successfully attacked (cf. subsection 5.4).

5.3.1 Confidentiality

The previous subsection discussed how a quantum-capable attacker could break both the authentication and encryption components of TLS. This subsection discusses more specifically how a quantum-capable adversary can compromise the confidentiality guarantees of classically secure protocols; subsection 5.3.2 discusses the implications to secure authentication.

5.3.1.1 Harvest-and-Decrypt

Knowing that it is intractable to decrypt ciphertext using classical computers, attackers may instead choose to capture the encrypted data and store it until they have a quantum computer; at which time the data can be decrypted. This practice is known as *harvest-and-decrypt* and it is well known that state actors around the globe are already doing it.

Harvest-and-decrypt does not only apply to TLS-secured Internet traffic, but to any kind of encrypted data an attacker may be able to get their hands on. Harvest-and-Decrypt is especially of concern for sensitive, high-value data that needs to be kept secret for a long time. For example, confidential healthcare data may need to be kept secure for as long as 50 years. This implies that sensitive data needs to be secured sooner rather than later with quantum-safe algorithms or, by hybrid classic/quantum algorithms (cf. subsection 5.3.3).

5.3.2 Authentication and Data Integrity

Classic authentication protocols tend to employ either RSA or ECDSA signatures; both of which are vulnerable to attack by Shor's algorithm. This implies that classically secure authentication algorithms need to be replaced and new quantum-safe key pairs need to be generated and distributed. Any hardware with embedded keys will be vulnerable to attack for the lifetime of those keys.

Digital signatures are not only used to authenticate nodes in a network. Another use of digital signatures is to check that data has not been tampered with or altered in any way; this is known as data integrity. Data integrity is crucial for many applications including: digital contracts, financial transactions, and record keeping. Again, this implies that sensitive data needs to be secured (signed) with quantum-safe cryptography well before quantum computers are available (e.g. 5 years).

Keyed-Hash Message Authentication Codes (HMACs) provides data integrity (as well as authenticity) and are already very common in the financial services industry. Moreover, quantum-safe HMAC constructions are available, and quantum security proofs have been demonstrated [13]. It follows that standardized HMAC schemes are a reasonable choice for a post-quantum data integrity solution (e.g. SHA2 or SHA3 based).

5.3.3 Interoperability and Time Required for Migration

Securing protocols and networks with quantum-safe algorithms is not a straightforward task. Due to the intricacy and size of these networks, upgrading them without introducing service interruptions, or while maintaining interoperability between nodes is difficult. Moreover, there is a lack of maturity in quantum-safe cryptography in general; more time and effort needs to be put into the cryptanalysis of the purportedly quantum-safe schemes. The consequence of this is that the security of these schemes is not yet as established as say, RSA.

As experience with AES, SHA2, and ECC has shown, it takes years for new algorithms to be adopted and implemented and for the old algorithms (DES, SHA1) to be retired. If applications, servers, browsers, etc., are not using quantum-safe algorithms by the time universal quantum computers are available, then they will be vulnerable to quantum attacks. This raises questions about when and how quantum-resistant cryptography should be standardized and implemented.

Hybrid approaches can help avoid or minimize the migration issues outlined above. That is, instead of replacing classic algorithms with quantum-resistant alternatives, algorithms whose security relies on hard classic problems as well as difficult quantum problems are introduced.

5.4 Impact to Enterprise PKI

PKI handles the creation and management of public-key certificates. They may support Virtual Private Networks (VPNs), secure e-mail (S/MIME), secure web browsing (HTTPS), remote key loading (RKL) and in general can be quite varied and offer complex functionality. Remote VPNs usually use TLS or IPsec to secure traffic, and so they are also susceptible to quantum attacks. In addition, secure e-mail protocols such as S/MIME also rely on classically secure key establishment methods. Hence, one concern for enterprise PKI is that classically secure components have to be upgraded to use quantum-safe cryptography.

Another concern is that applications and protocols used within enterprise PKI (and across the Internet) could potentially inherit security assumptions from layers below in addition to their own security considerations unless a well-designed and secure network topology is implemented. Such a topology will serve to protect above layers from having to deal with the same security considerations as lower layers. This consideration is of course not a new one, but nonetheless, the capabilities of a quantum-capable attacker should be considered at each layer in the model.

Other concerns for enterprises include secure code signing and over-the-air software updates. For example, if software updates are signed classically, then end users become vulnerable to malicious updates sent by attackers impersonating the authentic developers. The implication here is that over-the-air updates to secure software against quantum-capable attackers will need to be done before large-scale quantum computers are available. Likewise, any signed software that needs to remain authenticated needs to be resigned with quantum-safe signature schemes, and the new keys need to be distributed.

6 Quantum-Safe Options

There are five main branches of mathematics thought to be suitable for the development of quantum-safe cryptosystems. Cryptography based on those five fields does not in general need to be implemented on a quantum computer to be used; it can be run on classical computers. Quantum Key Distribution (QKD) however, is a physics-based quantum-safe method of establishing shared secret keys which does require specialized quantum-enabled hardware. The European Telecommunications Standards Institute (ETSI) published an in-depth comparison of promising quantum-safe key establishment procedures including QKD [25]. The document is especially useful as it includes discussion on security and implementation considerations.

This section serves as an introduction to each of the five quantum-safe areas of mathematics as well as QKD. Table 2 below lists all six quantum-safe options and summarizes the types of cryptographic protocols they are known to yield. A checkmark (✓) means that area is suitable for developing that type of system, a cross (X) means that it is not suitable, and a dash (-) means it is not yet known if that area is suitable.

Approach	Quantum-Safe Option	Digital Signature	Public-Key Encryption	Key Agreement
Physics	Quantum Key Distribution	X	X	✓
Mathematics	Hash Functions	✓	X	X
	Lattices	✓	✓	✓
	Error Correcting Codes	-	✓	✓
	Isogenies	-	✓	✓
	Multivariate Polynomials	✓	-	-

Table 2: Quantum-Safe Options

6.1 Quantum Key Distribution

Of the six options for quantum security presented in this section, Quantum Key Distribution (QKD) is unique in the sense that it harnesses the quantum mechanical properties of nature and is not based on any difficult math problems. QKD allows two parties to establish a secure and random shared secret key; there are no signature or encryption schemes in QKD. Below is a high-level description of the first QKD protocol (BB84 [14] and [15]) between two parties: Alice as the sender, and Bob as the receiver.

Firstly, Alice and Bob must have an authenticated line of communication between them. To achieve this, a quantum-safe authentication protocol or a pre-shared secret key is required. The channel between the communicating parties is usually fibre optic or a direct line of sight over the (potentially large) distance between Alice and Bob. The initiating party, Alice, encodes data in some way; traditionally this is done by polarizing photons (vertically or diagonally) and sends that data stream over the quantum channel. Due to the probabilistic nature of quantum mechanics, Bob will not be able to recover the original data exactly, and so Alice and Bob do not necessarily share identical data at this point.

Bob will then communicate with the sender over an unauthenticated public classical channel to determine which data were corrupted when reading the initial message. Both the sender and receiver will discard the corrupted data. The remaining data is now identical and can be used to derive shared keys. In other QKD schemes, Alice and Bob might apply a strongly universal hash function[4] to derive a shared seed for key derivation.

Alice and Bob need an authenticated communications channel between them to be sure they are communicating with who they think they are. However, there is no requirement that this channel be private. This is because Alice and Bob can very precisely calculate how much data (in an information theoretic sense) an eavesdropper, Eve, was able to ascertain. Avoiding the technical details, Alice and Bob can do this calculation using the quantum mechanical principal that it is impossible to measure a quantum state without collapsing it. More formally, Alice and Bob apply the no-cloning theorem [17].

[4] Which is like a hash function with information theoretic guarantees [16].

6.2 Hash-Based Cryptography

Hash-based signatures can be classified into three basic types: one-time, few-time, or N-time. One can further distinguish a signature scheme as either stateful or stateless. A signature scheme is a one-time scheme if it is secure when at most one signature is produced under that instance's signing key. Few-time signature (FTS) schemes are secure if "not too many" signatures are produced, and N-time schemes are secure if at most N signatures are produced. Stateful schemes are those that maintain internal states (or counters) that need to be incremented after each new signature is issued; stateless schemes do not have this requirement. The most mature and efficient hash-based signature schemes today are stateful.

A Merkle Tree [30] is a balanced, binary tree where each non-leaf node is the hash value of the concatenation of its children. At a very high-level, N-time signature schemes are constructed by composing many instances of one-time (or few-time) schemes together into a Merkle Tree. In those sorts of constructions, each leaf node corresponds to an instance of a one-time signature (OTS) scheme (or FTS scheme); specifically, the leaf nodes are calculated from the public-keys of those instances.

A complete signature for an N-time scheme includes at least one OTS or FTS and its associated *authentication path*; the ordered collection of sibling nodes in the tree required to compute the leaf-to-root path. This is required because, the global public-key for these schemes is the root node, and verification of a signature is done by computing a candidate root node to compare with the global public key. Since each leaf can only be used to sign at most once (or at most "a few times" in the case of FTSs), N-time schemes constructed like this are effectively 2^n-time schemes where n is the height of the Merkle tree. To increase the number of signatures even more, one may instantiate an N-time scheme into a hierarchical construction; this essentially involves treating an instance of the N-time scheme as a node in a yet larger virtual tree.

Figure 3 below gives an example of a signature and authentication path for a height 3 Merkle tree. The signing peer uses the secret key for the OTS instance (shown in blue) to produce a signature, and a verifying peer uses the corresponding verification key, and authentication path (shown in green) to construct a candidate root node. The signature is accepted if and only if the candidate root node equals the global public key.

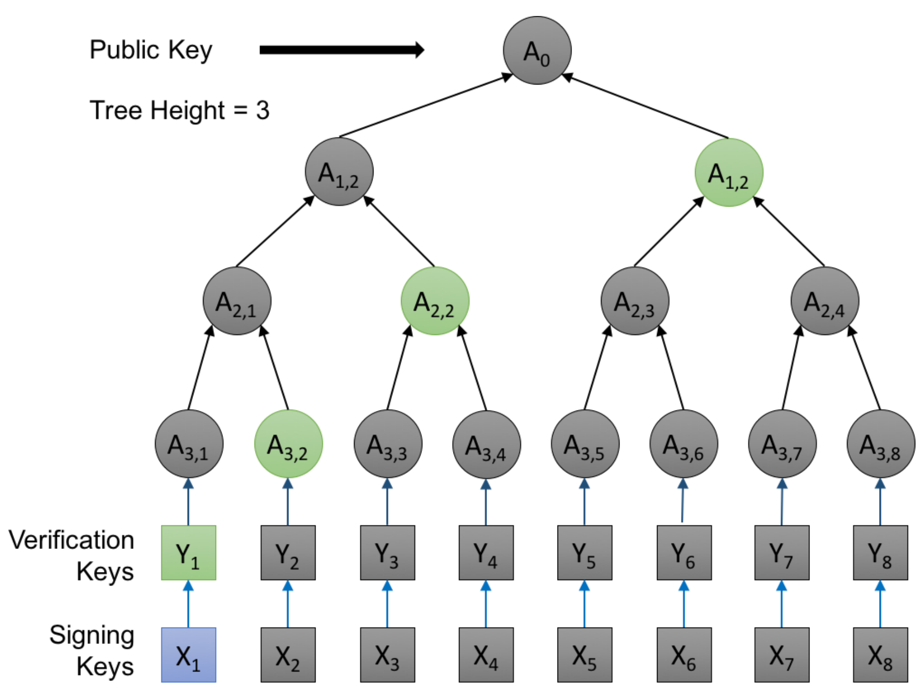

Figure 1: Merkle tree with Signature

The most promising stateful candidates for standardization are the Leighton-Micali Signature (LMS) scheme [39] [40], and the eXtended Merkle Signature Scheme (XMSS) [35] [36] [37]. The first semi-practical, stateless hash-based signature scheme was SPHINCS [41], but its signatures are too large for many applications (41kB), and so improvements and optimizations are currently being studied.

6.3 Code-Based Cryptography

At a high-level, code-based encryption schemes involve multiplying a plaintext vector by a public-key matrix and perturbing the product with an error vector. The weight of the error vector must be small enough so that the ciphertext can be decoded correctly and efficiently, but large enough to prevent an attacker from feasibly recovering the plaintext. Code-based encryption schemes tend to have relatively fast encryption and decryption, but large keys.

In general, the security of code-based cryptosystems is based on the difficulty of decoding random linear codes. The problem of decoding errors in a binary code is known to be NP-hard, and the decisional variant of the problem is known to be NP-complete [43].

Code-based encryptions schemes can easily be transformed into key establishment protocols. This has been done for example in the recent CAKE scheme [44]. Code-based signature schemes on the other hand, have been few, and the security and performance analyses for those schemes have been minimal; more research is required in this area.

The original code-based scheme by McEliece [42] (using binary Goppa codes) was proposed in 1978 and is still secure by today's standards but is impractical. The Quasi-Cyclic Moderate Density Parity Check (QC-MDPC) McEliece encryption scheme [45] was very promising, but a recent key-recovery attack prevents the scheme from being used in a static setting. Authors have successfully mitigated the key recovery attack by using ephemeral keys, but the fix changes the functionality of the scheme. There are quite a few code-based proposals out there, many of which use different types of codes such as: rank metric codes, Reed-Muller codes, (generalized) Srivastava codes, and so on. The security investigation of such schemes is ongoing.

6.4 Lattice-Based Cryptography

The theory of lattices is a leading contender for the development of quantum-safe cryptosystems. The popularity of lattice-based cryptography is due to several reasons. Firstly, there are more hard problems in lattice theory than in other areas; this leads to greater flexibility in the types of systems lattices can yield. Secondly, the mathematical theory of lattices has been studied for much longer than other candidate primitives, and so, there is more maturity in the abstract mathematics. Lattice-based systems are also attractive because they tend to offer very competitive performance and key sizes.

The most important hard problems on lattices are the Learning With Errors (LWE) and Small Integer Solution (SIS) problems. The former is similar to the decoding problem of Section 6.3, and the latter is in some sense the dual, or opposite of the LWE problem [47] [48]. These problems are particularly attractive because of Regev's average-to-worst-case reduction which showed that the LWE problem is on average as hard as other lattice problems in their worst cases [48].

A promising lattice-based key establishment protocol is Kyber (a Key Encapsulation Mechanism (KEM) that can also be used for encryption) [52], and a candidate lattice-based signature schemes is Dilithium [51]. It is worthwhile to note that 29 of the 69 "complete and proper" submissions to the NIST PQC standardization project were lattice-based [53].

6.5 Multivariate Polynomial-Based Cryptography

Introduced by Matsumoto and Imai in 1988 [55], multivariate cryptography is the branch of cryptography that uses composed systems of multivariate polynomials defined over a finite field to encrypt, or sign data. The security of multivariate schemes is based on the NP-hard problem of solving non-linear, multivariate polynomial systems. In fact, this problem remains NP-hard when restricted to the case where each polynomial in the system is quadratic. Multivariate quadratic polynomials are most often used in practice over higher degree systems because higher degree systems can often be reduced to quadratic systems, and the performance trade-offs of using higher degree systems favour quadratic polynomials.

While a small number of multivariate polynomial key establishment protocols have been proposed, they have not been well studied, and have generally been specialized to particular types of networks. Multivariate encryption schemes have been more popular than key establishment algorithms, with the most well-known being the Hidden Field Equations (HFE) and its variants: HFE+/-, HFEv+/-, ZHFE, MultiHFE, and so on. However, researchers remain sceptical of the security of these encryption schemes due to the plurality of schemes that have been broken or weakened by attacks such as Faugère's attacks on HFE using Gröbner bases or the Shamir-Kipnis key recovery attack.

Multivariate polynomials seem to be best suited for the construction of digital signature schemes. The advantage of multivariate signature schemes is that a signature is a solution to the publicly known system of equations. Hence, to verify a signature one need only evaluate the system on the signature. Examples of candidate quantum-safe multivariate signature schemes are: Rainbow [58], GUI [60], and HMFEv [59].

6.6 Isogeny-Based Cryptography

Isogeny-based cryptography is the youngest of the areas discussed in this section. The first known use of isogenies in cryptography was in a 1997 presentation by Couveignes, which was published in 2006 [64]. In 2006 Rostovtsev and Stolbunov proposed public-key cryptosystems based on isogenies defined between ordinary elliptic curves [63], but ordinary curves were shown to be weak against quantum-capable adversaries by Childs, Jao, and Soukharev in late 2010 (revised in early 2011) [65]. In 2011, Jao and De Feo proposed a quantum-safe key exchange protocol based on isogenies defined between *supersingular* elliptic curves. The Jao-De Feo key exchange scheme is known as Supersingular Isogeny Diffie-Hellman (SIDH) [61] [62].

The security of supersingular isogeny-based schemes relies on the difficulty of computing an isogeny between two supersingular elliptic curves. This problem is thought to be difficult even when it is known that such an isogeny exists. The problem has been studied for over two decades, starting with the work of Kohel [66]. The 2011 paper by Childs et al presented a subexponential-time quantum algorithm for the case of ordinary elliptic curves. To date, the fastest known quantum algorithm for the supersingular case takes exponential-time with subexponential memory requirements [67] and [69].

Isogeny-based cryptosystems are in general much slower than other quantum-safe options, but they have the smallest key sizes out of all currently known quantum-safe cryptosystems; which for many applications, is more important than speed. It should be noted that new research over the past few years has greatly improved the security, efficiency and key sizes of the SIDH scheme.

The main isogeny-based cryptosystem is SIDH, and the only known encryption schemes are derived from SIDH. A small number of supersingular isogeny-based signature schemes have been proposed, but none of them are considered practical [70] [71] [72] and [73]. The development of practical, quantum-safe, supersingular isogeny-based signature schemes is an active area of research.

7 Quantum Computers and CMS

The cryptographic message syntax (CMS) is designed to deliver the following services:

1) Independent data unit protection, where each message or transaction is protected independently. There is no need for a real-time communications session between sender and recipient, and no cryptographic sequencing (such as cipher block chaining) between messages.

2) Confidentiality, using symmetric encryption algorithms and key management algorithms. Typically, a key management algorithm is used with a Key Encryption Key (KEK) to protect a Content Encryption Key (CEK) that is used to encrypt the message. This approach allows the sender to send an encrypted message to multiple recipients, while only encrypting the actual message once with the CEK, then encrypting the CEK with a KEK for each recipient. The syntax is optimized for the common case where the same key management algorithm and parameters are used for all recipients.

3) Integrity and data origin authentication, using digital signature, MAC or HMAC algorithms. When digital signatures are used, non-repudiation may also be supported. The syntax supports multiple signers, per-signer authenticated attributes, unsigned attributes, and countersignatures.

4) Confidentiality, integrity, data origin authentication and non-repudiation, using signcryption algorithms. Signcryption mechanisms offer the capability of unforgeability, (i.e., the ability to detect data modifications, even modifications by a message recipient), a stronger notion of security than offered by symmetric authenticated encryption techniques.

Each of these services rely on the secure management and protection of the cryptographic keys involved. The following subsection discusses the impact quantum computers will have on the key management mechanisms of CMS.

7.1 Impact to Key Management in CMS

7.1.1 Impact to Authentication

Authentication of peers in CMS typically requires the use of digital signatures. As previously mentioned, the signature schemes used today (RSA and ECDSA) are susceptible to attack by Shor's algorithm. Thus, quantum-safe signature schemes will need to be deployed within the CMS to ensure secure authentication of peers in the presence of a quantum-capable attacker.

To prove message authenticity, Alice will first use her private signing key sk_A to generate a signature using a message digest (hash) of suitable output length (generally at least 256 bits). Next, she will send the message along with the signature and her certificate containing her public key pk_A (the public-key corresponding with sk_A) to Bob. Bob will verify Alice's certificate by verifying the digital signatures in the certificate chain. Finally, he will compute the message digest and verify the signature on it.

If Alice and Bob are using a classically secure signature algorithm, say RSA, then a quantum-capable attacker Eve may be able to recover sk_A from pk_A by factoring the RSA modulus with Shor's algorithm. The implication here is that Eve can successfully impersonate Alice by giving Bob a valid signature on her challenge text. Additionally, Eve can recover the private key of any CA certificate in the chain and forge all the certificates below the compromised certificate.

7.1.2 Impact to Key Establishment

X9.73 [2.1] defines mechanisms for conveying a symmetric key (for encryption or the computation of an authentication code) in a key management information structure. The mechanisms are: key transport, key agreement, symmetric key encryption key, password-based encryption, and other. This section investigates how a quantum-capable adversary impacts key transport and key agreement.

In a key transport protocol, a CEK is determined by the initiating party and encrypted under the public key of the recipient. The resulting ciphertext is sent to the receiving peer who can decrypt it using their private key and recover the CEK. At this point, the receiving peer can decrypt any content encrypted with the recovered CEK. A more concrete example is given below.

Suppose Alice wishes to encrypt a message and send it to Bob. Alice will generate a random symmetric, say AES, key K, and encrypt the message with this key. Using Bob's RSA public key pk_B, Alice will pad and encrypt K

using RSA (PKCS #1 v1.5) encryption to produce ciphertext c. Upon receiving c from Alice, Bob decrypts it with his RSA secret key sk_B to recover K. Both parties now share the secret AES key and Bob can decrypt the message Alice encrypted using that key.

In the above example, a eavesdropper Eve can harvest Bob's public key pk_B and recover his secret key sk_B—when she gains access to a quantum computer—by using Shor's algorithm. Such an attack would allow Eve to decrypt the AES key K for the current and any future communications with Bob; breaking any confidentiality

In a key agreement protocol, Alice fetches Bob's static public key agreement key pk_B (e.g. a Diffie-Hellman key in a certificate with the keyAgreement bit set in the key usage extension). This key is combined with Alice's (static or ephemeral) private key agreement key sk_A to create a shared secret S, which is passed through a KDF to generate the CEK. Alice sends her public key agreement key pk_A to Bob. Bob then combines pk_A with his own private key sk_B to generate S, which he runs through the same KDF to get the CEK.

If in the above example an eavesdropper Eve learns the values of either pk_A or pk_B, then she may be able to recover the corresponding sk_A or sk_B—when she gains access to a quantum computer—using Shor's algorithm and therefore determine the shared secret S, allowing her to decrypt future communications.

CMS allows a single message to be sent to multiple recipients in one package. To avoid encrypting the message multiple times, and thus drastically increasing the size of the package, CMS generates a single CEK for a package. That CEK is then encrypted individually for each recipient, adding only the overhead of an encrypted CEK per recipient rather than the full encrypted message per recipient.

Observe then that any encrypted content is only as secure as the least secure recipient. Even if most of the recipients are using quantum-safe or hybrid cryptography, the communications are still susceptible to attack by a quantum-capable attacker if even a single recipient is using classic cryptography exclusively.

7.1.3 Impact to Symmetric Key Encryption

As discussed in Section 5.2.3, Grover's algorithm roughly reduces the security of symmetric encryption schemes by half. Hence, to maintain current security levels, the key lengths of symmetric schemes will need to be roughly doubled.

7.2 General Recommendations for CMS

This section lays out some general recommendations for improving the security of CMS during the quantum-safe standardization and transition periods.

7.2.1 Recommendations for Digital Signatures

As mentioned in Section 5.3.3, migrating systems to new quantum-safe signature algorithms will be a long process with many nuances to be carefully considered and addressed. This process will be made somewhat easier once quantum-safe algorithms have been properly vetted and standardized. However, no suitable quantum-safe signature schemes have been standardized yet. The process of standardizing new signature schemes is underway and is not expected to produce any standards for some years yet. Exceptions to this are the stateful hash-based signature schemes XMSS and LMS. XMSS has been approved by the CFRG and published as the de factor standard RFC 8391 [38], and LMS is nearing RFC publication as well [40].

During this interim period before quantum-safe signature schemes are standardized, hybrid signing methods may be employed instead. That is, composing quantum-safe signature schemes with at least one traditional, classically secure scheme. While we may not have 100% confidence in the quantum-safe digital signature algorithms, using multiple quantum-safe options in addition to a single classic option provide some level of assurance over purely classical systems.

7.2.2 Recommendations for Key Establishment

Section 7.1.2 showed how current key transport and key agreement protocols are compromised by quantum-capable attackers. Key compromises are among the most devastating attacks within the realm of secure communications, and as such, key establishment protocols should be secured as soon as possible against harvest-and-decrypt attacks. As alluded to in other sections, hybrid approaches are a viable way of achieving this.

Many of the proposed solutions to this problem have been KEMs rather than public-key encryption schemes The following subsections discuss ways in which hybrid approaches and KEMs can be used to achieve quantum-safe key establishment in CMS.

Not provided in this document is any guidance on key bundling or distribution methods. However, it is worth mentioning that if these processes are not done in a quantum-safe manner then serious security concerns may arise. In those situations, best practices are needed.

7.2.2.1 Hybrid Methods

Hybrid schemes have distinct classically and quantum secure components. Hybrid cryptosystems can be made such that both of those components need to be broken for an attack against them to succeed. Hybrid constructions allow communicating parties to protect their communications both with classical and quantum-safe cryptography, and in some cases, allow them to encrypt their communications with multiple quantum-safe, classical, and symmetric systems.

During the transition period from classical to quantum-safe cryptography, confidence in purportedly quantum-safe cryptosystems will not be as high as most would prefer. This is simply a consequence of the schemes being relatively new. To increase confidence then, peers can negotiate shared secrets for multiple quantum-safe schemes, each of whose security relies on different problems than the rest. These secrets can be combined in some way (such as concatenation) and that result can be used to derive shared symmetric keys.

This sort of multiple-cryptosystem-negotiation is difficult to achieve in key transport protocols. One of the reasons for this is because many quantum-safe cryptosystems cannot encrypt arbitrary messages, but rather require a specific format of input. Thus, if a ciphertext is produced in one layer of encryption, it is not necessarily true that it can be further encrypted under a different scheme.

Rather than attempting to encrypt a CEK in multiple layers using different algorithms, each algorithm can either wrap or generate a portion of the seed which will be used to generate a key-encryption key (KEK). Algorithms that typically wrap a CEK can instead wrap a random value of the same size and use that as its seed. Thus, each algorithm, i, to be used in the hybrid construction will produce a seed s_i and ciphertext c_i. The seeds are concatenated and put through a KDF to produce the KEK, $kek = KDF(s_1, s_2, \dots)$ and the KEK is used as a symmetric key to encrypt the CEK, e.g. using AES. All of the ciphertexts c_i are concatenated along with the encrypted CEK and sent to the receiver who can decrypt the ciphertexts to recover the seeds and thus generate the KEK needed to decrypt the encrypted CEK. Each hybrid construction will depend on the classical and quantum-safe public keys available to use, and so the ordering of the concatenated ciphertexts will have to be done in some predictable way so that the receiver can correctly decrypt each ciphertext to recover the seeds.

As mentioned in Section 5.3.1.1, the practice of harvest-and-decrypt is already being exercised by actors around the world. Because of this, harvest-and-decrypt poses a greater threat to key establishment in CMS today than a quantum-capable attacker does (because there aren't any yet). Because of this, it seems reasonable to consider using hybrid key establishment approaches during this interim period, as have been considered in other secure protocols such as IPsec [77] and TLS [78]. Another option to increase quantum security is to establish Pre-Shared Keys (PSKs) for use as additional inputs to protocols. This has, for example, been proposed as a near-term solution in CMS [79].

7.2.2.2 KEMs in CMS

The quantum-safe key establishment algorithms proposed thus far have been KEMs. This is in large part due to the NIST post-quantum cryptography standardization project explicitly calling for KEMs (in addition to public-key encryption and digital signature algorithms) [74]. It is useful to note that KEMs have been proposed for use within the CMS before; such as with the IETF's RSA-KEM [75]. This section describes a general method for how key encapsulation could be used in CMS.

The following definition is a slightly modified version of that found in *A Designer's Guide to KEMs* [76]. The definition given below better reflects how KEMs are used in practice[5].

A KEM is a triple of algorithms:

- a key generation algorithm *KEM.Gen*, which takes as input a security parameter 1^λ, and outputs a public/private key-pair *(pk,sk)*;
- an encapsulation mechanism, *KEM.Encap*, that takes as input a public-key pk and a seed s and outputs an encapsulated key-pair *(K, C)*;
- a decapsulation algorithm, *KEM.Decap*, that takes as input an encapsulated key C and a private-key sk, and outputs a seed s.

One can think of a KEM as a specialized type of key transport protocol. A key encapsulation algorithm takes as input a seed and the receiving peer's public key, and outputs a key and a ciphertext. The ciphertext is such that when it is decrypted under the receiving peer's private key (the private key matching the public key used in *KEM.Encap*) it returns the seed that was input into the encapsulation algorithm by the sending peer.

Thus, after running *KEM.Encap,* the sending peer keeps the key and sends the ciphertext to the receiving peer. The receiving peer can then run the decapsulation algorithm to recover the seed needed to derive the shared key using the encapsulation algorithm. After this point, the key can be used as a CEK, KEK, or it can be used to derive further secrets.

In CMS, the encapsulation portion of a KEM scheme takes as input a receiver's public key, pk_B. pk_B can encrypt any number up to n. The KEM randomly generates a number, $m < n$. This avoids having to add any padding before encrypting. m is encrypted using pk_B to produce ciphertext c. m is also put through a KDF to produce a key-encrypting key kek. Ciphertext c and key kek are the outputs of the KEM.

In CMS, a content-encryption key, cek, is generated and used to encrypt the message, as in the usual key transport case. cek is encrypted using key kek to produce encrypted CEK cek_{enc}. The ciphertext and encrypted CEK are concatenated to produce the CMS encrypted keying data: $EK = c \,||\, cek_{enc}$. EK is sent to the receiver as part of the usual CMS message.

The receiver parses EK into c and cek_{enc}. As part of the KEM decapsulation, the receiver decrypts ciphertext c using its private key sk_B to recover m and puts m through a KDF to reproduce the key-encrypting key kek. cek_{enc}

[5] The definition from [76] does not take a seed as input into *Kem.Encap,* but rather generates randomness internally, and C decrypts to K instead of a seed. KEMs conforming to both versions have been proposed, and so both methods are mentioned in this report.

is decrypted using kek to recover cek, the original content-encryption key which is used to decrypt the message from the sender.

The KDF and symmetric algorithm used to encrypt the content-encryption key are defined as parameters of the KEM algorithm. Like a key transport algorithm, the KEM algorithm information is transmitted to the recipient as part of the usual CMS syntax.

7.2.3 Recommendations for Symmetric Key Encryption

For most applications, a security level of at least 128 bits is generally desired. As discussed previously, hash functions with *n*-bit output generally offer about *n/2*-bits of post-quantum security. The output lengths then should be increased to at least 256 bits so as to provide at least 128-bits of post-quantum security. The security levels of symmetric key cryptosystems correspond to the lengths of the keys used. And so, it is the key length that needs to be made longer to maintain security levels in those cases.

For concreteness, AES-128 provides 128 bits of classical security (against pre-image attacks), and about 64-bits of quantum security. To have at least 128-bits of security in a post-quantum setting then, AES-256 should be used. Similarly, hash functions like SHA-512 offer 256 bits of classical security and 128-bits of post-quantum security and hence, should be used instead of shorter output hash functions.

Bibliography

[1] M. Nielsen, and I. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge: Cambridge University Press, (2010) doi:10.1017/CBO9780511976667

[2] P. Kaye, R. Laflamme, and M. Mosca. *An Introduction to Quantum Computing*. Oxford University Press, Inc., New York, NY, USA, (2007).
– https://research.googleblog.com/2016/07/towards-exact-quantum-description-of.html

[3] M. Stenta, and M. Dal Peraro. *An introduction to quantum chemical methods applied to drug design*. In frontiers in bioscience, Vol E3, Iss. 1, pp. 1061-1078, (2011).

[4] D. B. Boyd. *Quantum mechanics in drug design: methods and applications*. In drug information journal. Vol 17, Iss. 3, pp. 121-131, (1983).

[5] Google AI Blog. *Towards an Exact (Quantum) Description of Chemistry*
– https://ai.googleblog.com/2016/07/towards-exact-quantum-description-of.html

[6] A. Marouchkine. *Room-Temperature Superconductivity*
– https://arxiv.org/ftp/cond-mat/papers/0606/0606187.pdf

[7] Science Alert. *Physicists Achieve Superconductivity at Room Temperature*
– https://www.sciencealert.com/physicists-achieve-superconductivity-at-room-temperature

[8] K. Tanaka, J. S. Tse, and H. Liu. *Electron-phonon coupling mechanisms for hydrogen-rich metals at high pressure.* In. Phys. Rev. B, Vol 96, (2017).
– https://journals.aps.org/prb/abstract/10.1103/PhysRevB.96.100502

[9] R. Mankowsky, A. Subedi, M. Först, S. O. Mariager, M. Chollet, H. T. Lemke, J. S. Robinson, J. M. Glownia, M. P. Minitti, A. Frano, M. Fechner, N. A. Spaldin, T. Loew, B. Keimer, A. Georges, and A. Cavalleri. *Nonlinear lattice dynamics as a basis for enchanced superconductivity in* $YBa_2Cu_3O_{6.5}$. In Nature, vol 516, pp. 71-73, (2014).

[10] U.S. Energy Information Administration: Frequently Asked Questions
– https://www.eia.gov/tools/faqs/faq.php?id=105&t=3

[11] M. Schuld, I. Sinayskiy, and F. Petruccione. *An introduction to quantum machine learning.* (2014).
– https://arxiv.org/abs/1409.3097

[12] T. A. Shaikh, and R. Ali. *Quantum Computing in Big Data Analytics: A Survey*. In. IEEE International Conference on Computer and Information Technology (CIT), pp. 112-115, (2016).
– https://ieeexplore.ieee.org/document/7876324/

[13] F. Song, and A. Yun. Quantum Security of NMAC and Related Constructions — PRF domain extension against quantum attacks. https://eprint.iacr.org/2017/509.pdf

[14] C. H. Bennett, and G. Brassard. *Quantum cryptography: Public key distribution and coin tossing.* In Proceedings of IEEE International Conference on Computers, Systems and Signal Processing, vol 175, pp. 8. New York, (1984).
– http://researcher.watson.ibm.com/researcher/files/us-bennetc/BB84highest.pdf

[15] C. H. Bennett, and G. Brassard. *Quantum cryptography: Public key distribution and coin tossing.* Theoretical Computer Science. Theoretical Aspects of Quantum Cryptography – celebrating 30 years of BB84. 560, Part 1: 7–11. doi:10.1016/j.tcs.2014.05.025. (2014).

[16] Wikipedia: Universal Hashing
– https://en.wikipedia.org/wiki/Universal_hashing.

[17] W. K. Wootters, and W. H. Zurek. *A Single Quantum Cannot be Cloned*. In Nature. Vol. 299, pp. 802-803. (1982). Bibcode:1982Natur.299..802W. doi:10.1038/299802a0.

[18] A. K. Lenstra, and H. W. Lenstra Jr. *The Development of the Number Field Sieve*. LNCS 1554. Springer-Verlag Berlin Heidelberg, (1993). https://www.springer.com/us/book/9783540570134/

[19] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. "Chapter 3" (PDF). Handbook of Applied Cryptography. (1997).

[20] P. W. Shor. *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*. SIAM J. Comput. 26, pp. 1484-1509, (1997).

[21] D. Beckman, A. N. Chari, S. Devabhaktuni, and J. Preskill. *Efficient networks for quantum factoring*. (1996). https://arxiv.org/abs/quant-ph/9602016

[22] L. K. Grover. *Quantum Mechanics Helps in Searching for a Needle in a Haystack.* Phys. Rev. Lett., Vol. 79, Iss. 2, pp. 325-328, (1997).

[23] P. Junod. *Linear Cryptanalysis of DES.* École polytechnique fédérale de Lausanne Diploma Thesis.
– https://www.apprendre-en-ligne.net/crypto/bibliotheque/PDF/lincrypt.pdf

[24] Quadibloc Differential and Linear Cryptanalysis
– http://www.quadibloc.com/crypto/co040501.htm

[25] ETSI TR 103 570: Cyber; Quantum-Safe Key Exchanges. (2017).
– https://www.etsi.org/deliver/etsi_tr/103500_103599/103570/01.01.01_60/tr_103570v010101p.pdf

[26] G. Brassard, P. Hyer, and A. Tapp. *Quantum algorithm for the collision problem*. ACM SIGACT News (Cryptology Column), Vol 28, pp. 14-19, (1997).

[27] D. J. Bernstein. *Cost analysis of hash collisions: Will quantum computers make SHARCS obsolete?* In SHARCS'09: Special-purpose Hardware for Attacking Cryptographic Systems, (2009).

[28] P. C. van Oorschot, and M. J. Wiener. *Parallel collision search with cryptanalytic applications*. Journal of Cryptology 12, pp. 1-28, (1999).

[29] F. T. Leighton, and S. Micali. *Large provably fast and secure digital signature schemes based on secure hash functions.* US Patent 5,432,852, July 11, (1995).

[30] C. H. Bennett, E. Bernstein, G. Brassard G, and U. Vazirani. *The strengths and weaknesses of quantum computation.* SIAM Journal on Computing. Vol. 26, Iss. 5, pp. 1510-1523, (1997) doi:10.1137/s0097539796300933.

[31] R. C. Merkle. *Secrecy, authentication, and public key systems*. PhD thesis, Department of Electrical Engineering, Stanford University, (1979).

[32] L. Lamport. *Constructing digital signatures from a one way function*. Technical Report SRI-CSL-98, SRI International Computer Science Laboratory, (1979).

[33] A. Hülsing. *Practical forward secure signatures using minimal security assumptions*. PhD thesis, TU Darmstadt, (2013).

[34] A. Hülsing, C. Busold, and J. Buchmann. *Forward secure signatures on smart cards*. In: L. R. Knudsen, and H. Wu. (eds.), SAC 2012. LNCS 7707, pp. 66-80. Springer, (2013).

[35] A. Hülsing. *W-OTS+ | Shorter signatures for hash-based signature schemes*. In: A. Youssef, A. Nitaj, A.E. Hassanien. https://eprint.iacr.org/2017/965

[36] A. Hülsing, L. Rausch, and J. Buchmann. *Optimal parameters for XMSS-MT*. In: A. Cuzzocrea, C. Kittl, D.E. Simos, E. Weippl, and L. Xu. (eds.) Security Engineering and Intelligence Informatics. LNCS 8128, pp. 194-208. Springer, (2013).

[37] A. Hülsing, J. Rijneveld, and F. Song. *Mitigating multi-target attacks in hash-based signatures*. In: C. Cheng, Ka. Chung, G. Persiano, and B. Yang (eds.) PKC 2016. LNCS 9614, pp. 387-416. Springer, (2016).

[38] A. Hülsing, D. Butin, S. Gazdag, J. Rijneveld, and A. Mohaisen. *RFC 8391 XMSS: Extended hash-based signatures.* https://datatracker.ietf.org/doc/rfc8391/

[39] J. Katz. *Analysis of a proposed hash-based signature standard.* In: L. Chen, D. McGrew and C. Mitchell. (eds.) SSR 2016, LNCS 10074, pp. 261-273. Springer, (2016).

[40] D. McGrew, M. Curcio, and S. Fluhrer. *Hash-Based Signatures*. Internet-Draft draft-mcgrew-hash-sigs-12, August 3, (2018). Work in progress. https://datatracker.ietf.org/doc/draft-mcgrew-hash-sigs/

[41] D. J. Bernstein, D. Hopwood, A. Hülsing, T. Lange, R. Niederhagen, L. Papachristodoulou, P. Schwabe, and Z. Wilcox-O'Hearn. *SPHINCS: practical stateless hash-based signatures.* In M. Fischlin and E. Oswald, (eds.), Advances in Cryptology. EUROCRYPT 2015. LNCS 9056, pp. 368-397. Springer, (2015).

[42] R. J. McEliece. *A public-key cryptosystem based on algebraic coding theory.* In Jet Propulsion Laboratory DSN Progress Report pp. 42–44, 114–116, (1978)
– http://ipnpr.jpl.nasa.gov/progress_report2/42-44/44N.PDF

[43] E. R. Berlekamp, R. J. McEliece and H. C. Van Tilborg. *On the inherent intractibility of certain coding problems.* IEEE Transactions on Information Theory, Vol. 24, Iss. 3, pp. 384-386, (1978).

[44] P. S. L. M. Barreto, S. Gueron, T. Güneysu, R. Misoczki, E. Persichetti, N. Sendrier, and J. Tillich. CAKE: Code-based Algorithm for Key Encapsulation
– https://eprint.iacr.org/2017/757.pdf

[45] R. Misoczki, J. Tillich, N. Sendrier, and P. S. L. M. Barreto. *MDPC-McEliece: New McEliece Variants from Moderate Density Parity-Check Codes.*
– https://eprint.iacr.org/2012/409

[46] D. J. Bernstein, T. Chou, and P. Schwabe. *McBits: fast constant-time code-based cryptography*
– http://binary.cr.yp.to/mcbits-20130616.pdf

[47] O. Regev. *The learning with errors problem*. http://www.cims.nyu.edu/~regev/papers/lwesurvey.pdf

[48] O. Regev. *On lattices, learning with errors, random linear codes, and cryptography*. Journal of the ACM, Vol. 56, Iss. 6, (2009). Preliminary version in STOC'05

[49] S. Chatterjee, A. J. Menezes, and P. Sarkar. *Another Look at Tightness.*
– https://eprint.iacr.org/2011/442.pdf

[50] S. Chatterjee, N. Koblitz, A. J. Menezes, and P. Sarkar. *Another Look at Tightness II*
– https://eprint.iacr.org/2016/360.pdf

[51] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé. *CRYSTALS – Dilithium: Digital Signatures from Module Lattices.* https://eprint.iacr.org/2017/633.pdf

[52] J. Bos, L. Ducas, E. Kiltz, T. Lapoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé. *CRYSTALS – Kyber: a CCA-secure module-lattice-based KEM.* https://eprint.iacr.org/2017/634.pdf

[53] Post-Quantum Cryptography Lounge: https://www.safecrypto.eu/pqclounge/

[54] M. R. Garey, and D. S. Johnson. *A guide to the theory of NP-completeness.* Computers and intractability, New York, (1979).

[55] T. Matsumoto, and H. Imai. *Public quadratic polynominal tuples for efficient signature-verification and message-encryption.* In Eurocrypt, vol 88, pp. 419-453. Springer, (1988).

[56] J. Patarin. *Cryptanalysis of the Matsumoto and Imai public key scheme of eurocrypt'88*. In Crypto, vol 95, pp. 248-261. Springer, (1995).

[57] O. Billet, and J. Ding. *Gröbner Bases, Coding, and Cryptography: Overview of cryptanalysis techniques in multivariate public key cryptography*. (2009)

[58] A. Petzoldt, S. Bulygin, and J. Buchmann. *Selecting parameters for the rainbow signature scheme.* Post-Quantum Cryptography, pp. 218-240, (2010).

[59] A. Petzoldt, M. Chen, J. Ding, and B. Yang. *HMFEv – and efficient multivariate signature scheme.* (2017) http://ws680.nist.gov/publication/get_pdf.cfm?pub_id=922939

[60] J. Ding, M. Chen, A. Petzoldt, and B. Yang. *Gui: Revisiting Multivariate Digital Signature Schemes based on HFEV-.* https://csrc.nist.gov/csrc/media/events/workshop-on-cybersecurity-in-a-post-quantum-world/documents/presentations/session1-ding-jintai.pdf

[61] D. Jao and L. De Feo. *Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies.* Post-Quantum Cryptography (2011).

[62] L. De Feo, D. Jao, and J. Plût. *Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies.* In J. Math. Cryptol. Vol. **8** Iss. 3, pp. 209-247, (2014).

[63] A. Rostovtsev, and A. Stolbunov. *Public-key cryptosystems based on isogenies.* https://eprint.iacr.org/2006/145.pdf. 2006.

[64] J. Couveignes. *Hard homogeneous spaces.* https://eprint.iacr.org/2006/291.pdf. 2006.

[65] A. M. Childs, D. Jao, and V. Soukharev. *Constructing elliptic curve isogenies in subexponential time*. https://arxiv.org/abs/1012.4019. (2011).

[66] D. Kohel, *Endomorphism rings of elliptic curves over finite fields.* PhD thesis of the University of California at Berkeley, (1996).

[67] J. Biasse, D. Jao, and A. Sankar. *A quantum algorithm for computing supersingular isogenies between elliptic curves.* In: W. Meier, and D. Mukhopadhyay (eds) Progress in Cryptology -- INDOCRYPT 2014. LNCS vol 8885. Springer, (2014).

[68] E. Fujisaki, and T. Okamoto. *Secure integration of asymmetric and symmetric encryption schemes.* In Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology, CRYPTO'99, pp. 537-554, London, UK. Springer-Verlag, (1999).

[69] S. D. Galbrait, C. Petit, B. Shani, and Y. B. T. *On the security of supersingular isogeny cryptosystems*. In: J. H. Cheon, and T. Takagi, (eds.) ASIACRYPT 2016. LNCS, vol. 10031, pp. 63-91. Springer, Heidelberg (2016). doi:10.1007/978-3-662-53887-6_3.

[70] D. Jao, and V. Soukharev. *Isogeny-Based Quantum-Resistant Undeniable Signatures*. In: M. Mosca (ed) Post-Quantum Cryptography. PQCrypto 2014. LNCS, vol 8772. Springer, Cham, (2014).

[71] Y. Yoo, R. Azarderakhsh, A. Jalai, D. Jao, and V. Soukharev. A post-quantum digital signature scheme based on supersingular isogenies. https://eprint.iacr.org/2017/186.pdf

[72] M. S. Srinath, and V. Chandrasekaran. *Isogeny-based quantum-resistant undeniable blind signature schemes* https://eprint.iacr.org/2016/148.pdf

[73] X. Sun, H. Tian, and Y. Wang. *Toward quantum-resistant strong designated verifier signature*. Int. J. Grid Util. Comput. Vol. 5, Iss. 2, pp. 80-86, (2014). DOI=http://dx.doi.org/10.1504/IJGUC.2014.060187

[74] NIST Post-Quantum Cryptography Standardization Project Call for Proposals. https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf

[75] J. Randall, B. Kaliski, J. Brainard, and S. Turner. *Use of the RSA-KEM Key Transport Algorithm in the Cryptographic Message Syntax (CMS)*. (2010) https://tools.ietf.org/html/rfc5990/

[76] A. W. Dent. *A designer's guide to KEMs*. In K. G. Paterson, ed, 9th IMA International Conference on Cryptography and Coding, LNCS 2898, pp. 133-151, Springer-Verlag, (2003).

[77] S. Fluhrer, D. McGrew, P. Kampanakis, and V. Smyslov. *Post-quantum Preshared Keys for IKEv2.* Work in progress. https://datatracker.ietf.org/doc/draft-fluhrer-qr-ikev2/

[78] W. Whyte, Z. Zhang, S. Fluhrer, and O. Garcia-Morchon. *Quantum-Safe Hybrid (QSH) Key Exchange for Transport Layer Security (TLS) version 1.3.* Work in progress. https://datatracker.ietf.org/doc/draft-whyte-qsh-tls13/

[79] R. Housley. Using Pre-Shared Key (PSK) in the Cryptographic Message Syntax (CMS). Work in progress (2018). https://datatracker.ietf.org/doc/draft-housley-cms-mix-with-psk/

Annex A

(Informative)

Quantum-Safe SignedData Structures

A.1 Countersignatures

Anticipated attacks on current digital signature algorithms using quantum computers will subject documents signed today and requiring long term protection to increasing security risk over time. This problem is not limited solely to quantum computing risks, such as risk of repudiation by the signer. Long termed signed documents such as a thirty-year mortgage are subject to similar risks.

These risks arise as new attacks on signature algorithms are discovered and key length requirements grow with computing power advances. An increase in the number of attacks, the continuing rise in legal and regulatory risks, and changes to the security polices of organizations all add to these risks. It is possible to mitigate some of these security risks using a quantum-safe countersignature over signed content created using current digital signature algorithms.

A countersignature that relies on a quantum-safe signature can be implemented in a `SignedData` message using an optional signed attribute. The schema of a `SignedData` message supports a series of signers represented in a value of type `SignerInfos`. Each signer is represented in this series as a value of type `SignerInfo`.

`Type SignerInfo is defined in the X9.73 standard as follows:`

```
SignerInfo ::= SEQUENCE {

   version              CMSVersion,

   sid                  SignerIdentifier,

   digestAlgorithm      DigestAlgorithmIdentifier,

   signedAttrs          [0] SignedAttributes  OPTIONAL,

   signatureAlgorithm   SignatureAlgorithmIdentifier,

   signature            SignatureValue,

   unsignedAttrs        [1] UnsignedAttributes  OPTIONAL

}
```

Type `SignerInfo` allows each signer to use a different signing key, message digest, and signature algorithm. Each signer can include their own set of attributes that will be cryptographically bound under their signature. A `signerInfos` attribute collects this series of values into an attribute that can be included in the signed attributes of a counter signer.

A `signerInfos` attribute is defined in X9.73 as follows:

```
signerInfos ATTRIBUTE ::= {
```



```
   WITH SYNTAX SignerInfos ID id-SignerInfos
}

id-SignerInfos OBJECT IDENTIFIER ::= { debs signerInfos(1) }
```

The `signerInfos` attribute contains a value of type `SignerInfos`, a series of values of type `SignerInfo`, one value for each signer of the `SignedData` content. Each cosigner shall sign content using their choice of signature and message digest algorithm. The signing key of each cosigner shall be included in the `SignerInfo` value of the cosigner using any of the choice alternatives defined in the X9.73 standard for signing key identification.

The optional `signedAttrs` component of type `SignerInfo` shall be present in the message. At a minimum, each cosigner shall include a `contentType` attribute and a `messageDigest` attribute in the `signedAttrs` component of their `SignerInfo` value. Additional signed attributes of any type or format may also be included by each cosigner. Any number or type of unsigned attributes may also be included by each cosigner in the `unsignedAttrs` component of their `SignerInfo` value.

The values of type `SignerInfo` created by all signers shall follow the defined processing steps and other requirements specified in the X9.73 standard. During signature verification of a `SignedData` message, a relying party may treat the `signerInfos` attribute as an opaque string. Applications that recognize this attribute may choose to defer signature verification processing. Failure of one or more cosigner `SignerInfo` values shall be handled as defined by the application.

A.2 Detached Content

When the detached form of `SignedData` is used, the `Content` component of the `SignedData` type is not present in the message. This message content must be available during signing and signature verification operations so that a message digest of the signed content can be calculated.

When default content location is not known to the communicating parties, content signers can include a `contentLocation` attribute in their signed attributes. This attribute can also be used when it is necessary for a cosigner to indicate a different detached object, such as a language-specific version of a contract.

A `contentLocation` attribute is defined as follows:

```
contentLocation ATTRIBUTE ::= {
   WITH SYNTAX URI ID id-ContentLocation
}

URI::= UTF8String (SIZE(1..MAX))

id-ContentLocation OBJECT IDENTIFIER ::= { debs contentLocation(2) }
```

A value of type `URI` is a Uniform Resource Identifier (URI) value that points to a location of detached `SignedData` content. A `contentLocation` attribute can be included in a `SignedAttributes` component of a `SignerInfo` component of type `SignedData` of any signer of the message. In some applications, it may be convenient to include a single content location attribute in the signed attributes of the counter signer.

A.3 Timestamp Considerations

Time stamps included in `SignedData` can be used to demonstrate that the validity period of a signer certificate included the time of signing a message. Long term signatures may need to be verified after the validity period of a signing certificate has expired. A time stamp attribute that is included in the `SignedAttributes` component can be compared by a relying party to the validity period of the signer certificate to ensure the certificate was valid for use when the message was signed.

A `timeStamped` attribute is defined as follows:

```
timeStamped ATTRIBUTE ::= {

   WITH SYNTAX TimeStamped ID id-TimeStamped

}

TimeStamped ::= SEQUENCE {

   timeStampValue    TimeStamp,

   timeStampService  URI  OPTIONAL

}

id-TimeStamped OBJECT IDENTIFIER ::= { debs timeStamped(3) }
```

Type `TimeStamped` contains two components, a required `timeStampValue` and an optional `timeStampService` that indicates the location of a time stamp service provider that can validate the time stamp. A `timeStampValue` component is a value of type `TimeStamp`, a choice between two alternatives, an X9.95 trusted timestamp token or a value from a local time source.

Type `TimeStamp` is defined in the X9.84 standard as follows:

```
TimeStamp ::= CHOICE {

   TimeStampToken    TimeStampToken, -- X9.95 Trusted Time Stamp --

   localTimeStamp    GeneralizedTime

}
```

The first choice alternative of type `TimeStamp` may be any of the four types of tokens defined in the X9.95 standard.