



KubeCon



CloudNativeCon

North America 2024

Running Quantum-Safe Applications on Kubernetes

Paul Schweigert & Michael (Max)imilien, IBM
psschwei.com @maximilien

Agenda

1. Understand the Risk
2. Becoming Quantum Safe
3. Protecting Applications
4. Next Steps

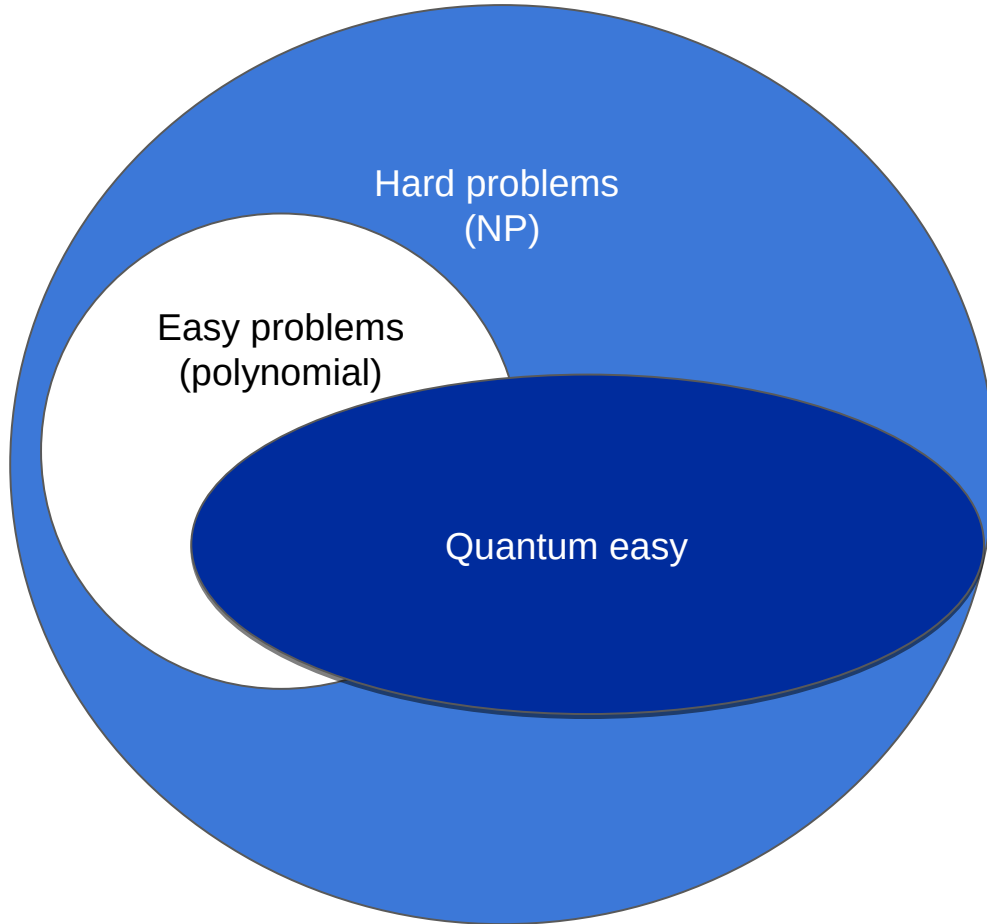
1. Understand the Risk

2. Becoming Quantum Safe

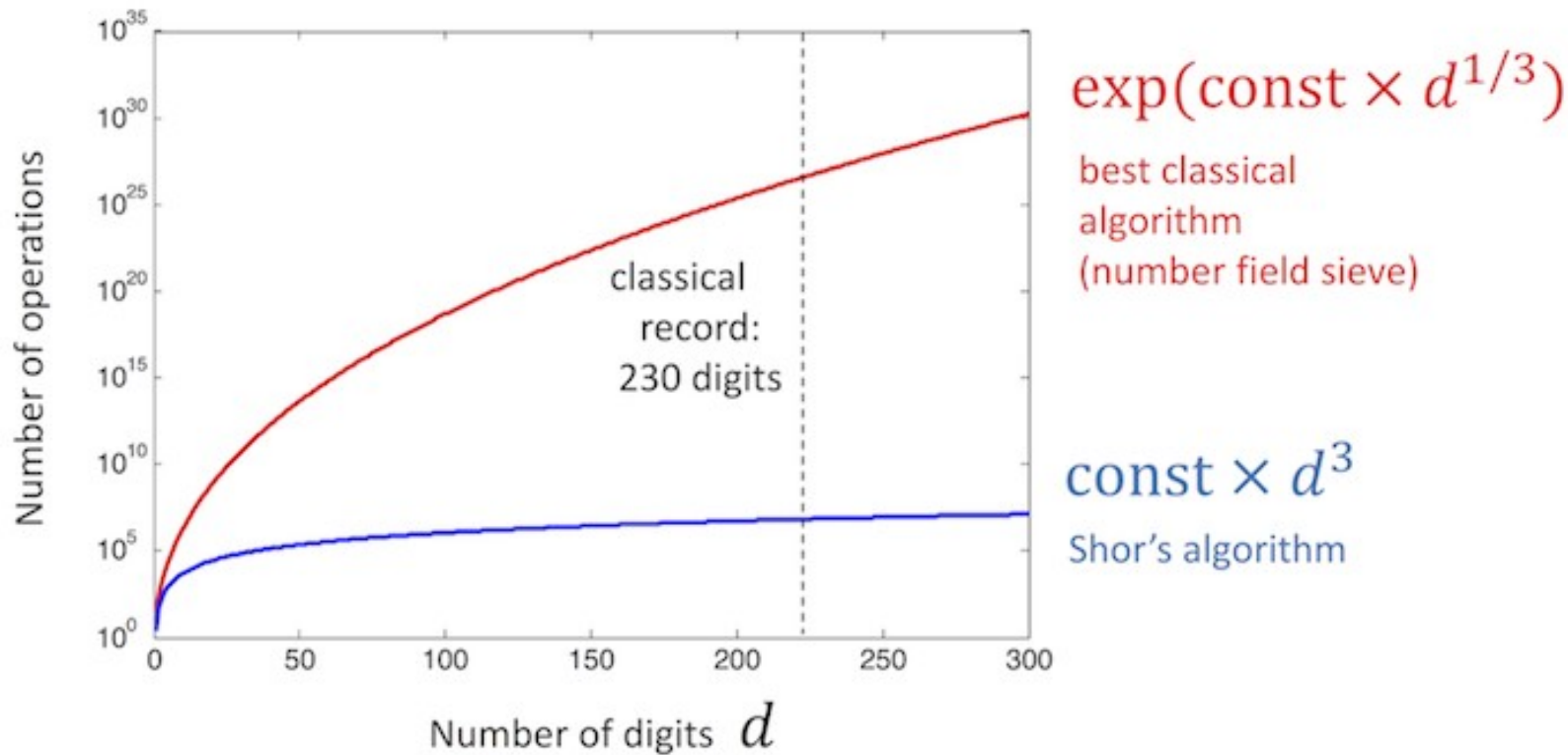
3. Protecting Applications

4. Next Steps

Why quantum?



Ex: Shor's algorithm for factoring



Current cryptography is at risk



Prime factors

$$= p \times q$$

2048-bit composite integer

```
251959084756578934940271832400483985714292821262040320
277771378360436620207075955562640185258807844069182906
412495150821892985591491761845028084891200728449926873
928072877767359714183472702618963750149718246911650776
133798590957000973304597488084284017974291006424586918
171951187461215151726546322822168699875491824224336372
590851418654620435767984233871847744479207399342365848
238242811981638150106748104516603773060562016196762561
338441436038339044149526344321901146575444541784240209
246165157233507787077498171257724679629263863563732899
121548314381678998850404453640235273819513786365643921
2010397122822120720357
```

Expected computation time

The most powerful computer **today:**

Millions of years

Shor's quantum algorithm:

Hours

Per Shor's algorithm, all public key crypto standards are vulnerable to attacks from large scale quantum computers

Public Key Encryption
Digital Signatures
Key Exchange Algorithms

RSA
DSA, ECDSA
Diffie-Hellman, ECDH

What will a cybercriminal be able to do?



Fraudulent
authentication



Forge digital
signatures



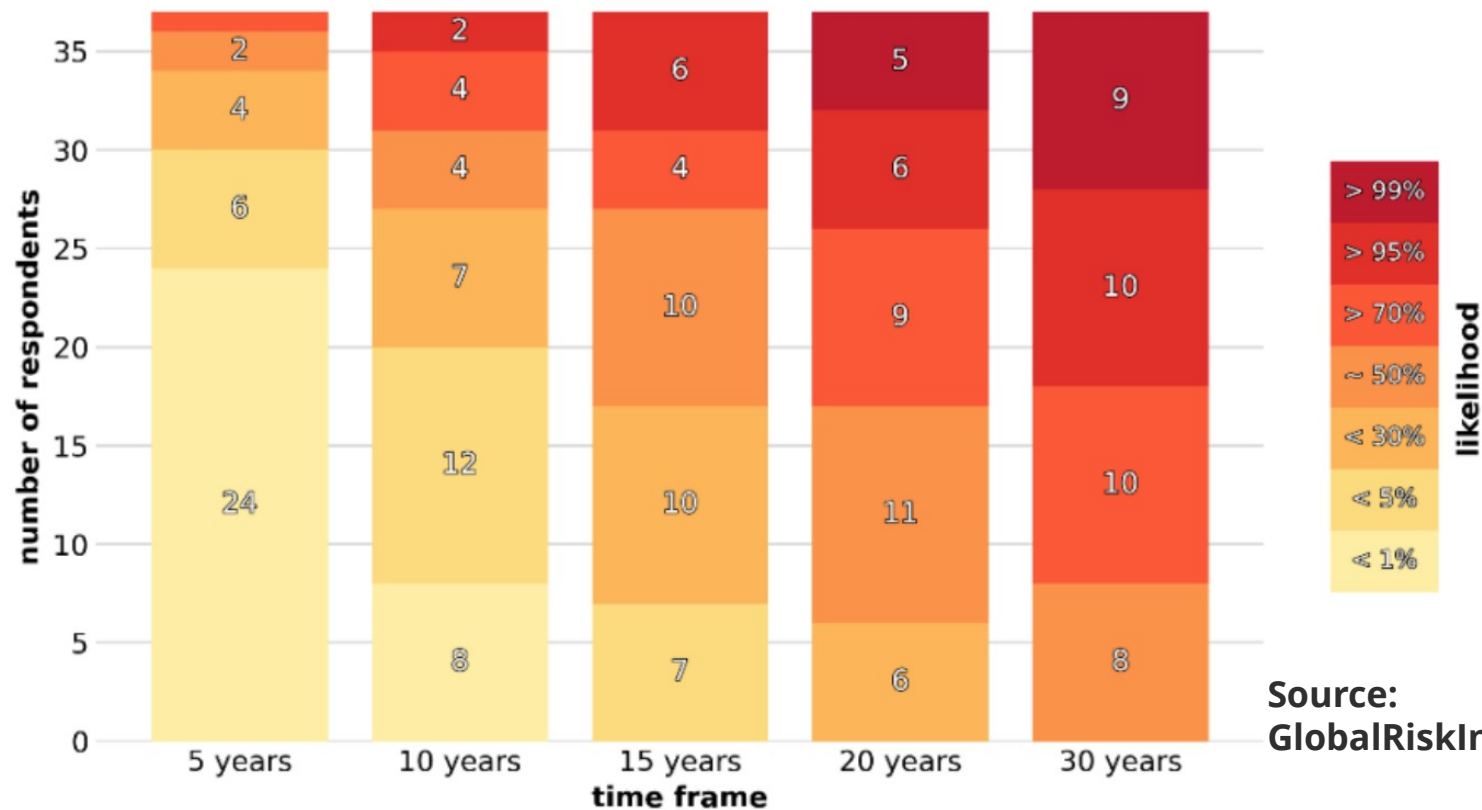
Harvest now,
decrypt later





2023 EXPERTS' ESTIMATES OF LIKELIHOOD OF A QUANTUM COMPUTER ABLE TO BREAK RSA-2048 IN 24 HOURS

Number of experts who indicated a certain likelihood in each indicated timeframe



Source:
GlobalRiskInsitute.org

1. Understand the Risk

2. Becoming Quantum Safe

3. Protecting Applications

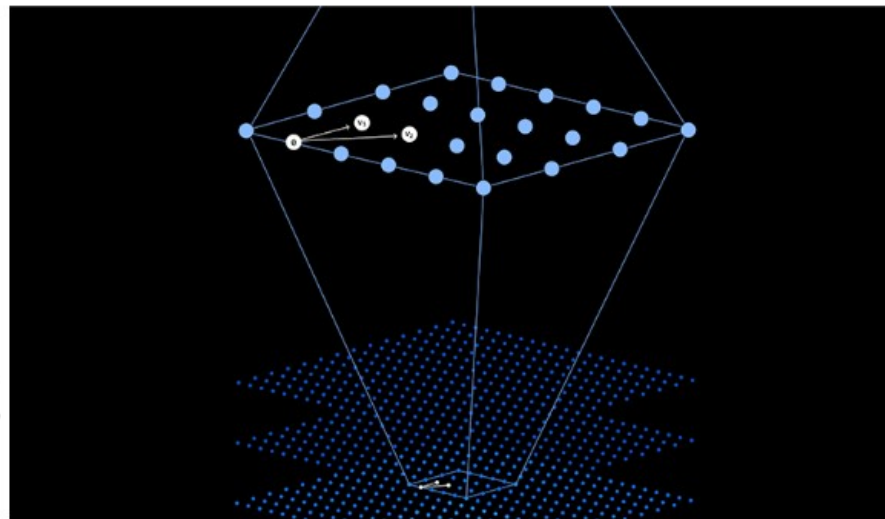
4. Next Steps

Quantum Safe Cryptography

a.k.a. Post Quantum Cryptography or Quantum Resistant Cryptography

Traditional public-key cryptography relies upon mathematical problems that are difficult to solve on classical computers.

Quantum-safe cryptography includes a suite of algorithms and systems that are resistant to attacks by both classical and quantum computers.



PROJECTS

Post-Quantum Cryptography PQC



Overview

Short URL: <https://www.nist.gov/pqcrypto>

[FIPS 203](#), [FIPS 204](#) and [FIPS 205](#), which specify algorithms derived from CRYSTALS-Dilithium, CRYSTALS-KYBER and SPHINCS⁺, were published August 13, 2024.

[4th Round KEMs](#)[Additional Digital Signature Schemes - Round 1 Submissions](#)[PQC License Summary & Excerpts](#)

For a plain-language introduction to post-quantum cryptography, go to: [What Is Post-Quantum Cryptography?](#)

Background

NIST initiated a process to solicit, evaluate, and standardize one or more quantum-resistant public-key cryptographic algorithms. **Full details can be found in the [Post-Quantum Cryptography Standardization](#) page.**

In recent years, there has been a substantial amount of research on quantum computers – machines that exploit quantum mechanical phenomena to solve mathematical problems that are difficult or intractable for conventional computers. If large-scale quantum computers are ever built, they will be able to break many of the public-key cryptosystems currently in use. This would seriously compromise the confidentiality and integrity of digital communications on the Internet and elsewhere. The goal of post-quantum cryptography (also called quantum-resistant cryptography) is to develop cryptographic systems that are secure against

PROJECT LINKS

Overview

FAQs

News & Updates

Events

Publications

Presentations

ADDITIONAL PAGES

Post-Quantum Cryptography Standardization

[Call for Proposals](#)[Example Files](#)[Round 1 Submissions](#)[Round 2 Submissions](#)[Round 3 Submissions](#)[Round 3 Seminars](#)

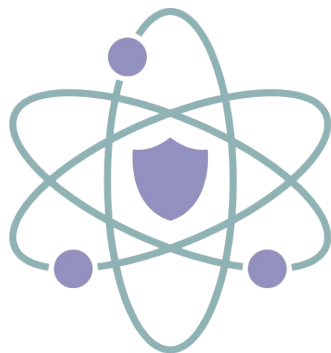
Round 4 Submissions

Selected Algorithms 2022

Workshops and Timeline

[PQC Seminars](#)

Open Source



**Post-Quantum
Cryptography Alliance**

<https://pqca.org>

To advance the adoption of post-quantum cryptography, by producing high-assurance software implementations of standardized algorithms, and supporting the continued development and standardization of new post-quantum algorithms with software for evaluation and prototyping.

Initial Projects Overview

Open Quantum Safe project

liboqs

Library of many PQ algorithms

- Main profile: standards-track algorithms
- Experimental profile: new algorithms, NIST signatures on-ramp etc.

OQS demos

Prototype integrations of PQ into protocols and applications to support experiments, standardization, interoperability

OQS OpenSSL 3 Provider

Integration of PQ + hybrid algorithms from liboqs into OpenSSL 3 via OpenSSL provider interface

- TLS key exchange, authentication
- X.509
- S/MIME, CMS, CMP

PQ Code Package

“Kyber” code package

High-assurance production source-code implementations of Kyber

- C, x86_64, ARMv8, ...
- Rust, Go, ...
- audited/certified/formally verified

Plus appropriate wrappers / providers, e.g. Kyber OpenSSL 3 provider

Potential Phase 2 projects

- Dilithium
- XMSS, LMS
- SPHINCS+
- Falcon (-> Phase 3?)

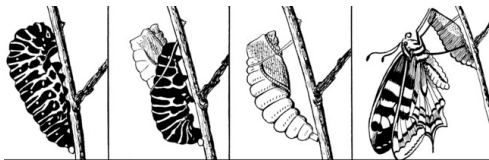
Becoming Quantum Safe



Discover: Scan source and object code to locate cryptographic assets, dependencies, and vulnerabilities. Build a cryptography bill of materials (CBOM).



Observe: Create a dynamic cryptographic inventory to guide remediation. Analyze cryptographic posture and compliance to prioritize risks.



Transform: Learn and apply quantum-safe remediation patterns in a development environment. Prepare to deploy quantum-safe solutions to your stack.

1. Understand the Risk
2. Becoming Quantum Safe
- 3. Protecting Applications**
4. Next Steps

Quantum Research

↳ Blog

Bringing quantum-safe security to IBM Quantum Platform, and the world

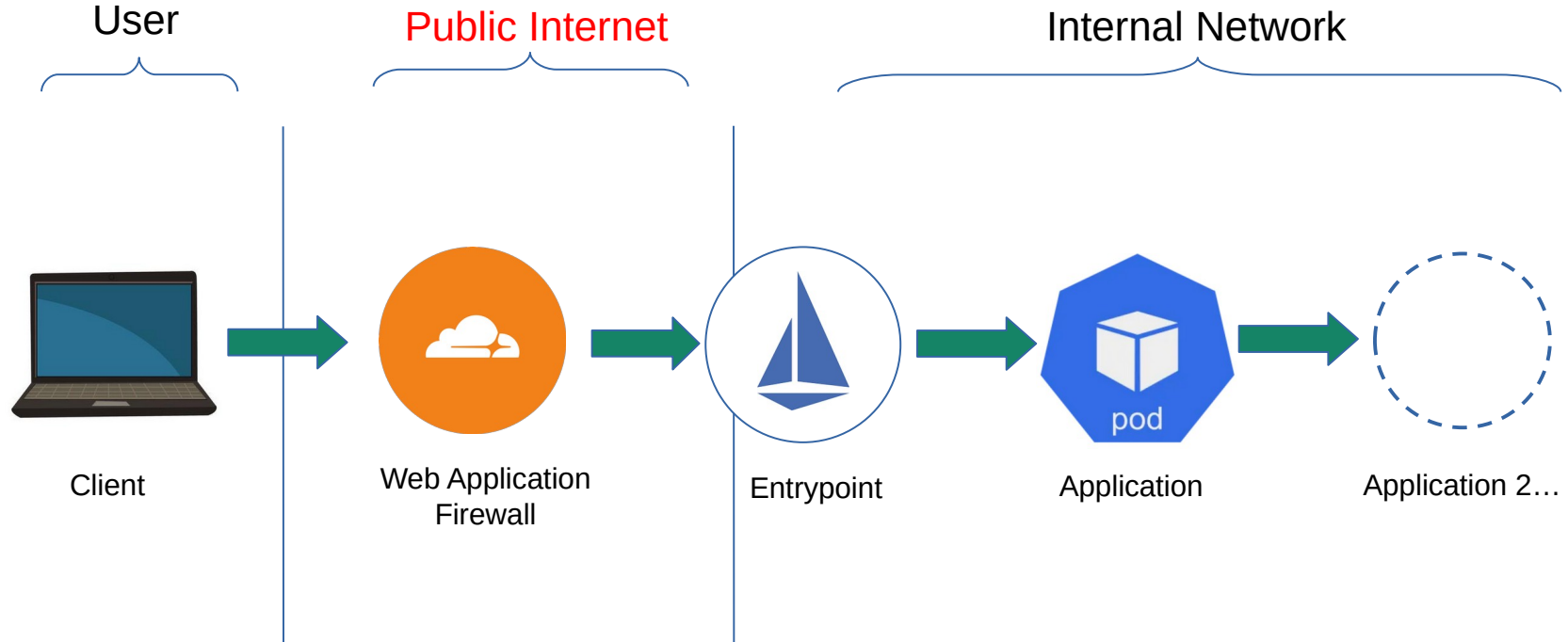
How IBM® is working to make itself, the open source community, and the world quantum safe.



<https://www.ibm.com/quantum/blog/iqp-quantum-safe>

Quantum Safe Flow

High Level View



```
graph TD; Client(client) -- "1. TLS call" --> OpenSSLCore[OpenSSL Core]; subgraph "OpenSSL Provider Stack"; OpenSSLCore -.-> "2. Request encryption algorithm" --> OQSProvider[OQS Provider]; OQSProvider -.-> "3. Get algorithm implementation" --> liboqs[liboqs library]; end
```

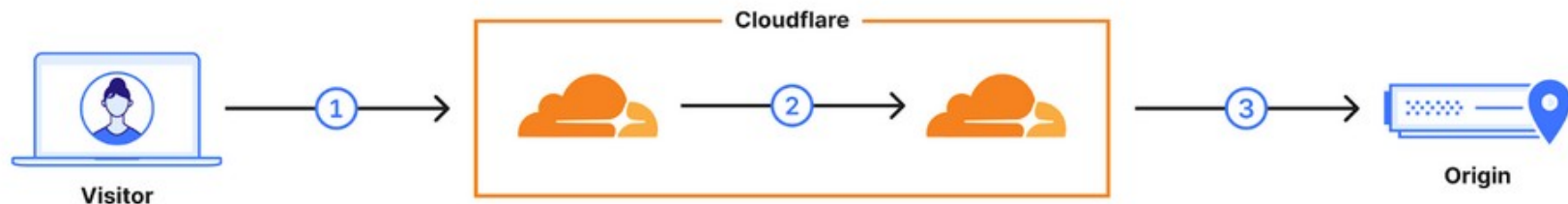
The diagram illustrates the OpenSSL Provider Stack architecture. A client initiates a TLS call to the OpenSSL Core. The OpenSSL Core then requests an encryption algorithm from the OQS Provider. The OQS Provider then gets the algorithm implementation from the liboqs library. The OpenSSL Core, OQS Provider, and liboqs library are all part of the OpenSSL Provider Stack.

- ```
78
79 [oqsprovider_sect]
80 activate = 1
81 module = /opt/conda/lib/openssl-modules/oqsprovider.so
82
83 [ssl_module]
84 system_default = tls_system_default
85
86 [tls_system_default]
87 TLS.MinProtocol = TLSv1.2
88 TLS.MaxProtocol = TLSv1.3
89 DTLS.MinProtocol = DTLSv1.2
90 DTLS.MaxProtocol = DTLSv1.2
91 Groups = x25519_kyber768
92 Ciphersuites = TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POL
93 CipherString = ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA
94
95 #####
96 [ca]
```

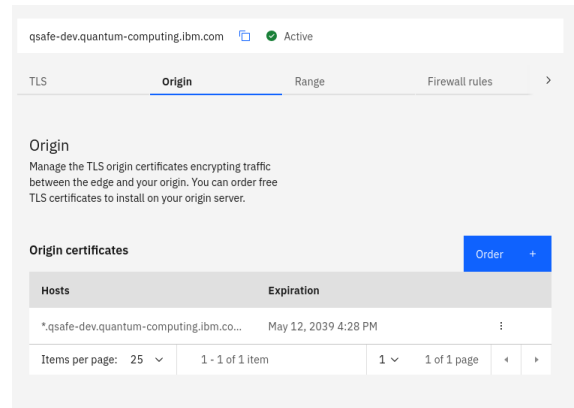
```
78
79 [oqsprovider_sect]
80 activate = 1
81 module = /opt/conda/lib/oqs-modules/oqsprovider.so
82
83 [ssl_module]
84 system_default = tls_system_default
85
86 [tls_system_default]
87 TLS.MinProtocol = TLSv1.2
88 TLS.MaxProtocol = TLSv1.3
89 DTLS.MinProtocol = DTLSv1.2
90 DTLS.MaxProtocol = DTLSv1.2
91 Groups = x25519_kyber768
92 Ciphersuites = TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POL
93 CipherString = ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA
94
95 #####
96 [ca]
```

# Quantum Safe Firewall:

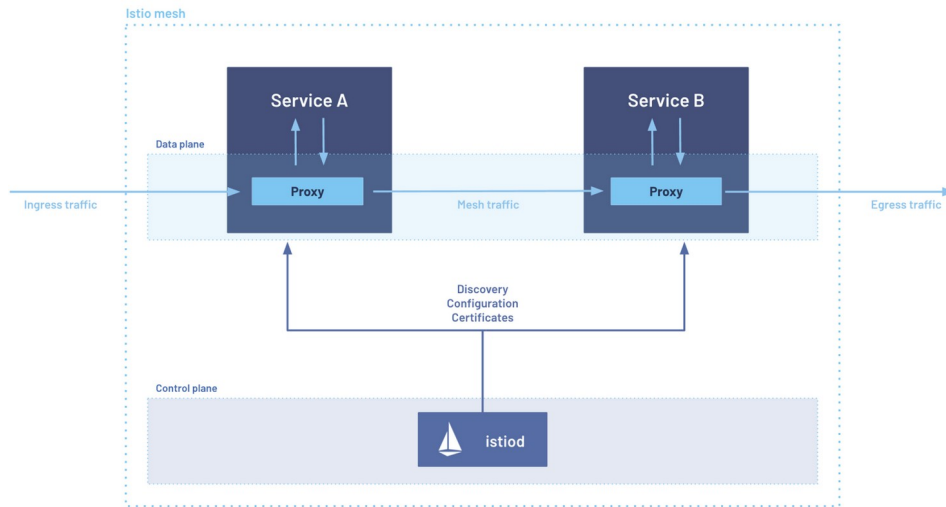
## Enabling PQC in Web Application Firewall



- Enable PQ encryption on IBM Cloud Internet Services
  - <https://cloud.ibm.com/apidocs/cis?code=go#update-origin-post-quantum-encryption>
- Create new origin cert for Ingress / VirtualService



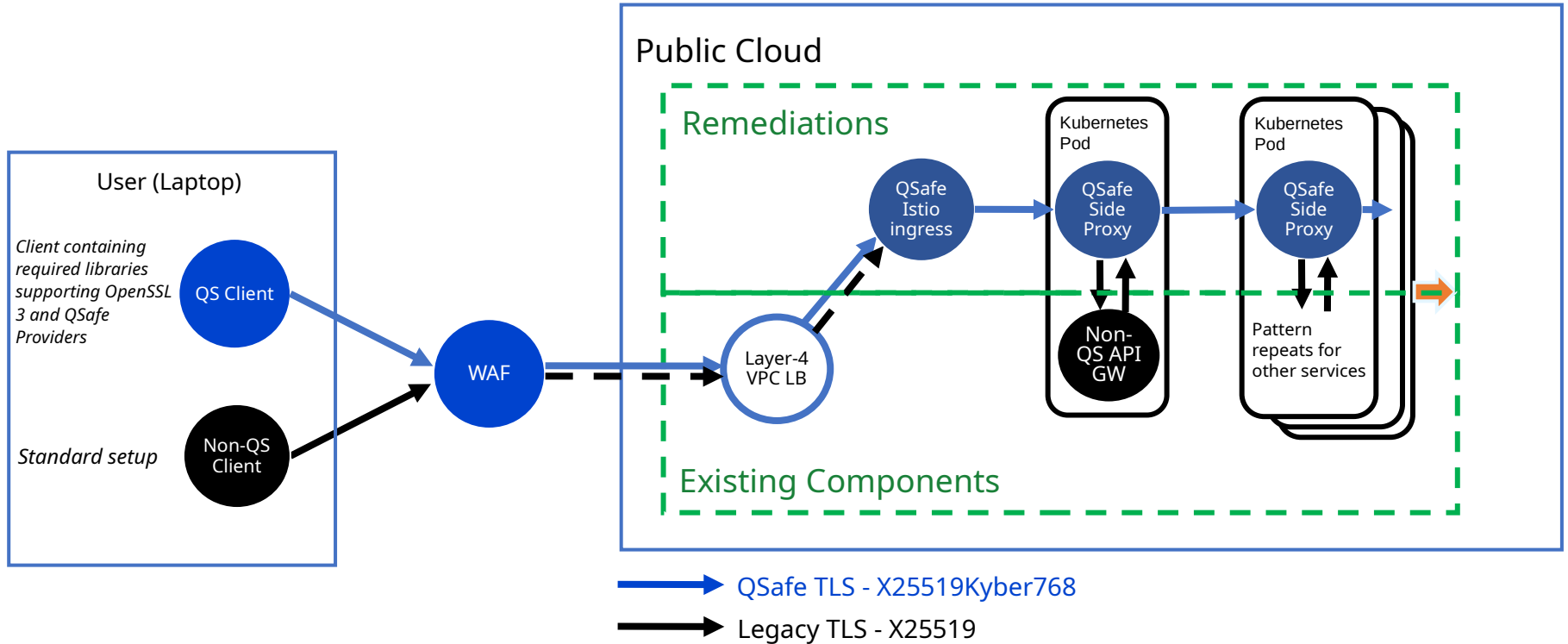
# Quantum Safe Service Mesh: Updating Istio



- Envoy
  - QSafe BoringSSL:  
[https://github.com/google/boringssl/blob/45cf810dbdbd767f09f8cb0b0fccccd342c39041f/src/ssl/ssl\\_key\\_share.cc#L285-L293](https://github.com/google/boringssl/blob/45cf810dbdbd767f09f8cb0b0fccccd342c39041f/src/ssl/ssl_key_share.cc#L285-L293)
- Istio
  - Add QSafe supported group: <https://github.com/istio/istio/commit/7635f7ea50514958518eb17b631682f953e723cc>
  - Secure mesh traffic: <https://github.com/istio/istio/issues/52290>

# Quantum Safe Flow

## Detail View





# Demo

1. Understand the Risk
2. Becoming Quantum Safe
3. Protecting Applications
- 4. Next Steps**

# Next Steps

## Learn about post-quantum cryptography

IBM Quantum Learning | Home | Catalog | Network | Composer | Lab

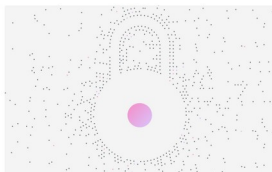
### Practical introduction to quantum-safe cryptography

Practical introduction to quantum-safe cryptography is a free online course on the IBM Quantum Learning platform and serves as a primer on the foundational concepts in quantum-safe cryptography.

This course was created for developers who are interested in modernizing their application security and features multimodal lessons and interactive live code examples in four core areas of cryptography: cryptographic hash functions, symmetric key cryptography, asymmetric key cryptography, and quantum-safe cryptography.

By taking this course, developers will learn how the cybersecurity risk landscape is evolving as well as explore contemporary approaches poised to shape the quantum era.

Start from beginning →



Lessons (0/7 complete)

Expand all lessons

Introduction to this course

- Course introduction
- Key takeaways for this course
- Tips for navigating this course
- Before you begin
- Lesson structure
- Running the Python examples
- Digital Certification
- Pre-course Survey
- Next steps after this course

Go to lesson →

Cryptographic hash functions

Symmetric key cryptography

Asymmetric key cryptography

Quantum-safe cryptography

What's next?

Works cited

Exam

Take this exam to test your skills. This exam is intended to be taken after reading the lessons in this course. Once you have completed the exam, come back here to see your earned badge.

You have passed the exam

Awarded badge



IBM leverages the services of Credly, a 3rd party data processor authorized by IBM and located in the United States, to assist in the administration of the IBM Digital Badge program. In order to issue you an IBM Digital Badge, your personal information (name, email address, and badge earned) will be shared with Credly.

You will receive an email notification from Credly with instructions for claiming the badge. Your personal information is used to issue your badge and for program reporting and operational purposes. IBM may share the personal information collected with IBM

## Start inventorying your crypto

```
"bomFormat": "CycloneDX",
"specVersion": "1.4-cbom-1.0",
"serialNumber": "urn:uuid:ebfc1f3c-0fbd-4803-9436-f46adbeccf6b",
"version": 1,
"metadata": {
 "timestamp": "2024-01-29T16:14:50.428825",
 "tools": [
 {
 "vendor": "IBM",
 "name": "Crypto_Scanner",
 "version": "1.0"
 }
],
 "component": {
 "type": "file",
 "bom-ref": "mycbom:94f37cf0276f46adbc2067b2ae687a8a6e7f1a7",
 "name": "mycbom",
 "version": ""
 }
},
"components": [
 {
 "type": "crypto-asset",
 "bom-ref": "CryptoUtils.java@CryptoUtils.java@12de7913-7d75-4641-bc46-ae9e40ee419",
 "name": "CryptoUtils.java",
 "version": "",
 "cryptoProperties": {
 "assetType": "algorithm",
 "algorithmProperties": {
 "primitive": "blockcipher",
 "variant": "AES-128-ECB",
 "mode": "ecb"
 },
 "relatedCryptoMaterialProperties": {},
 "classicalSecurityLevel": 128,
 "oid": "2.16.840.1.103.3.4.1.1",
 "scanner": "Crypto scanner version 0.8",
 "detectionContext": {
 {
 "filePath": "CryptoUtils.java@CryptoUtils.java@af7f74b5-f328-412f-824d-231f5c1d6b52",
 "lineNumbers": [
 105
]
 }
]
 }
 },
 {
 "type": "crypto-asset",
 "bom-ref": "BouncyCastleCrypto.java@BouncyCastleCrypto.java@7bd57bed-8d23-4341-bbf0-5a18d2b86bd8",
 "name": "BouncyCastleCrypto.java",
 "version": "",
 "cryptoProperties": {
 "assetType": "algorithm",
 "algorithmProperties": {
 "primitive": "blockcipher",
 "variant": "AES-128-ECB",
 "mode": "ecb"
 },
 "relatedCryptoMaterialProperties": {},
 "classicalSecurityLevel": 128,
 "oid": "2.16.840.1.103.3.4.1.1",
 "scanner": "Crypto scanner version 0.8",
 "detectionContext": {
 {
 "filePath": "BouncyCastleCrypto.java@BouncyCastleCrypto.java@7bd57bed-8d23-4341-bbf0-5a18d2b86bd8",
 "lineNumbers": [
 105
]
 }
]
 }
 }
]
```



# Next Steps

Learn about post-  
quantum cryptography



Inventory your crypto



Rate this session



Paul Schweigert  
psschwei.com  
paulschw@us.ibm.com

Michael (Max)imilien  
@maximilien  
maxim@us.ibm.com



**KubeCon**



**CloudNativeCon**

**North America 2024**

