

Crypto Agility in OpenSSL

Ryan Hooper

Norman Ashley



September 15, 2025

AGILITY



Agility refers to the ability to move the body quickly and easily, often involving rapid changes in direction or velocity in response to a stimulus. It encompasses both physical and mental aspects.



Courtesy of CIRCUIT



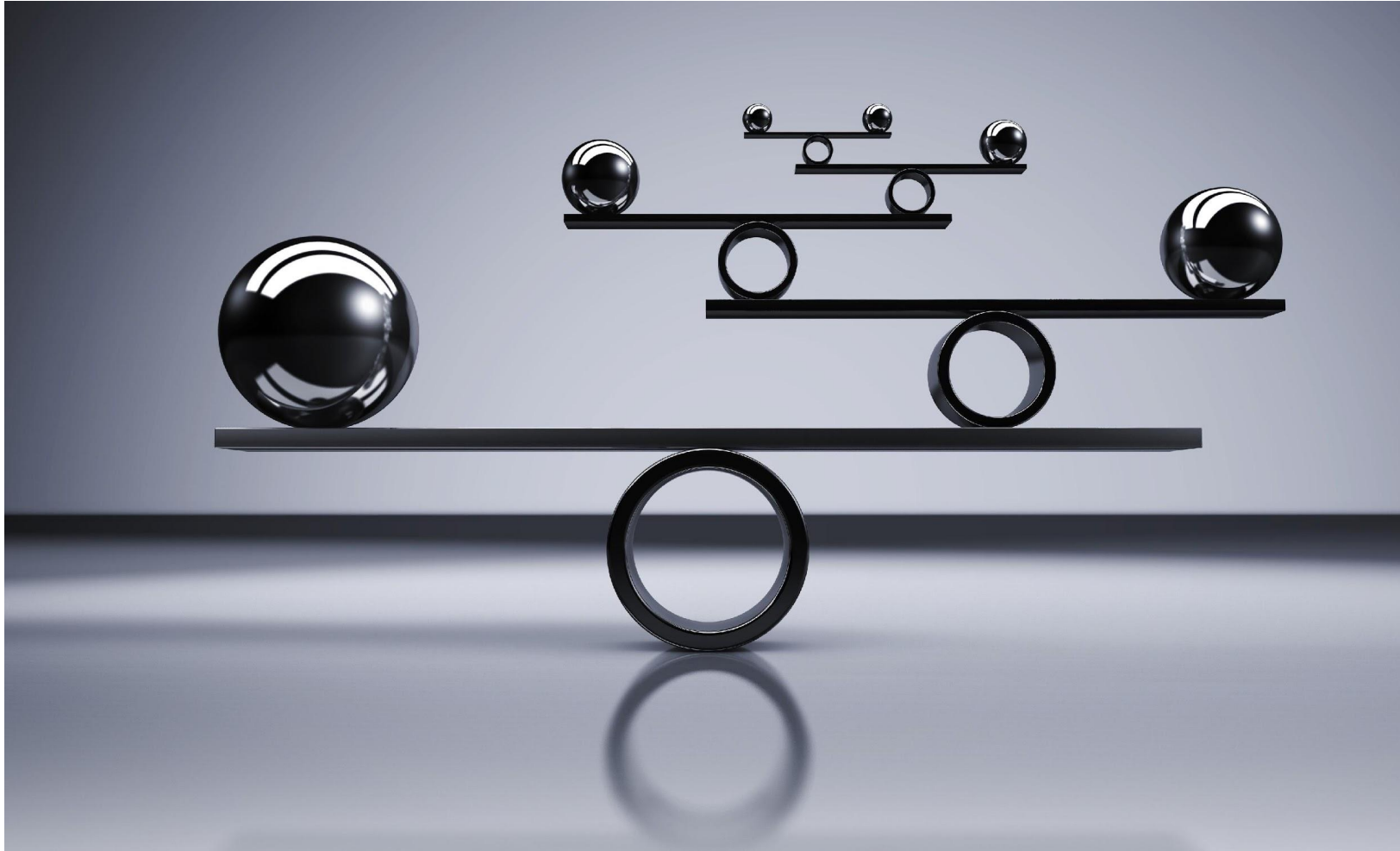
Crypto agility refers to the capacity of an information system or security system to swiftly and efficiently switch out cryptographic primitives, algorithms, and other encryption mechanisms without causing significant disruption to its infrastructure.

This capability is vital for maintaining data and system security in a constantly evolving threat landscape.



Courtesy of CIRCUIT

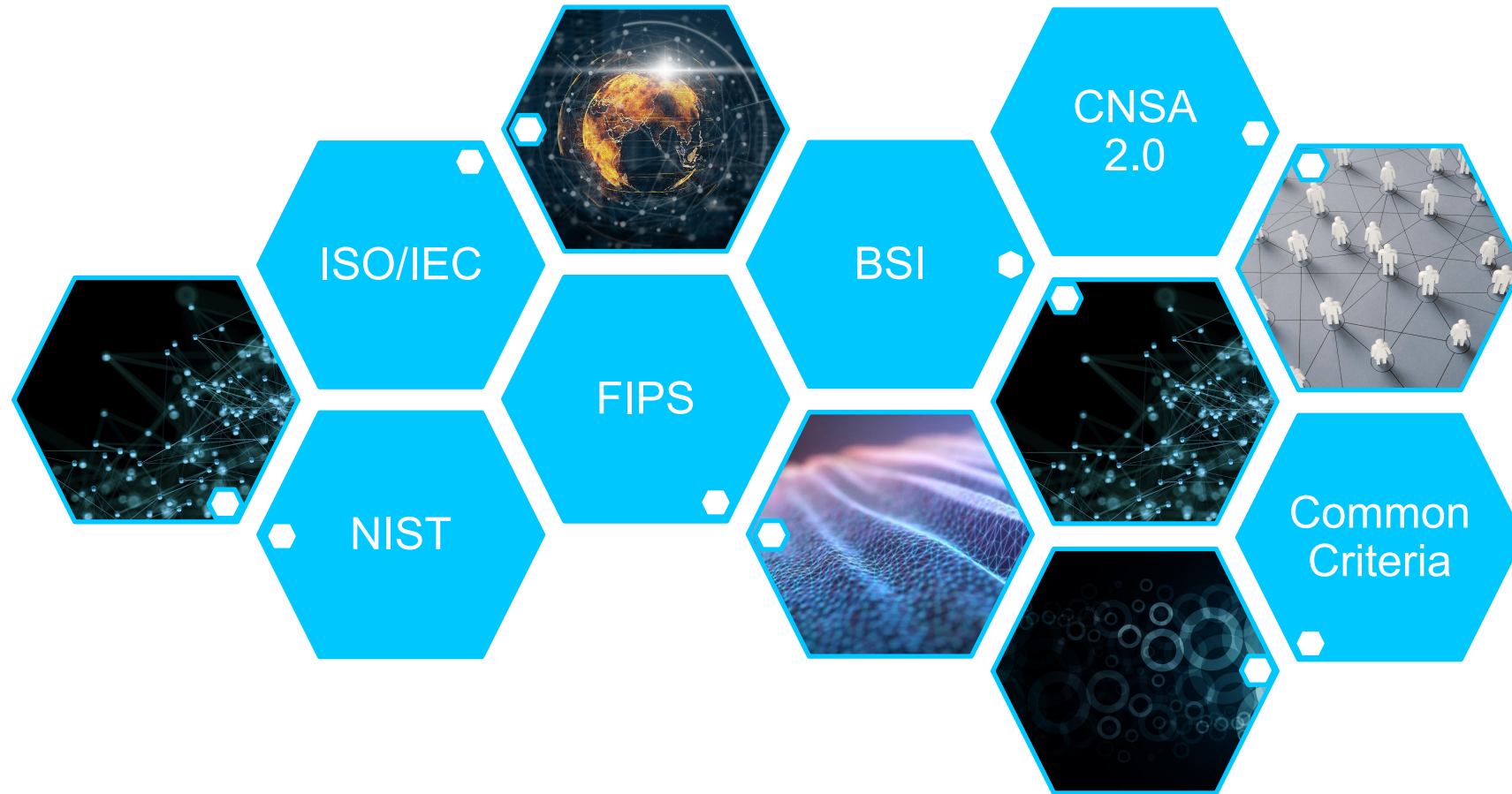
Crypto Agility



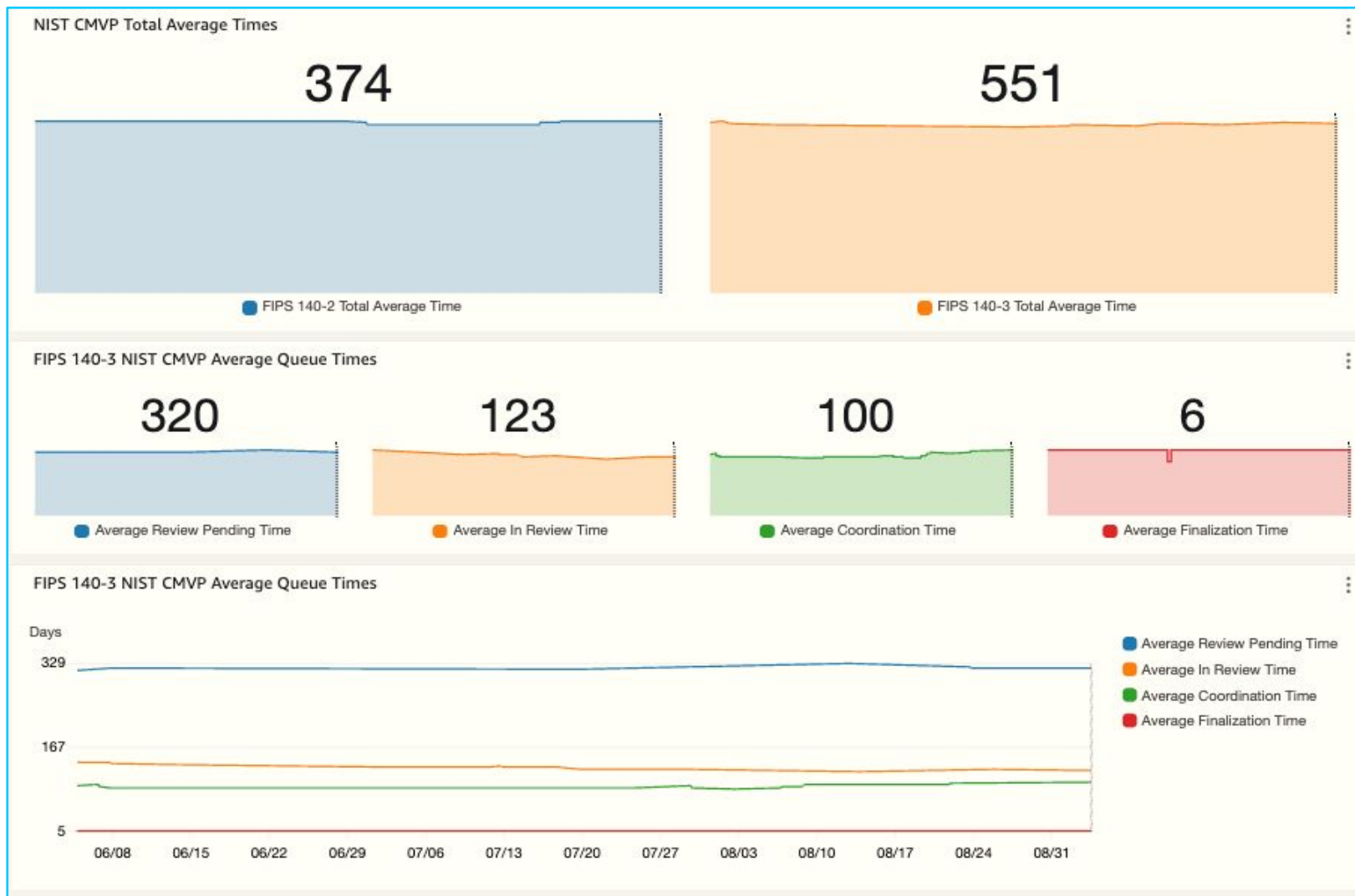
Agility



Governance Landscape



NIST CMVP Validation Queue Time

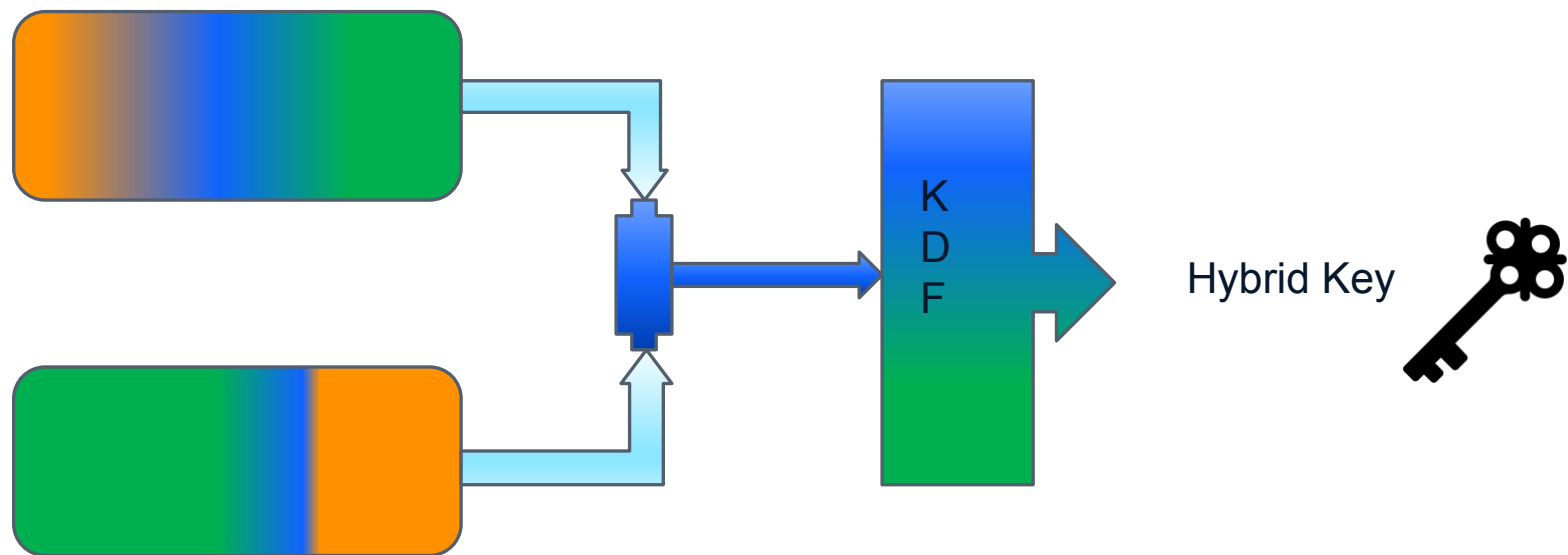


Courtesy of Alicia Squires/AWS

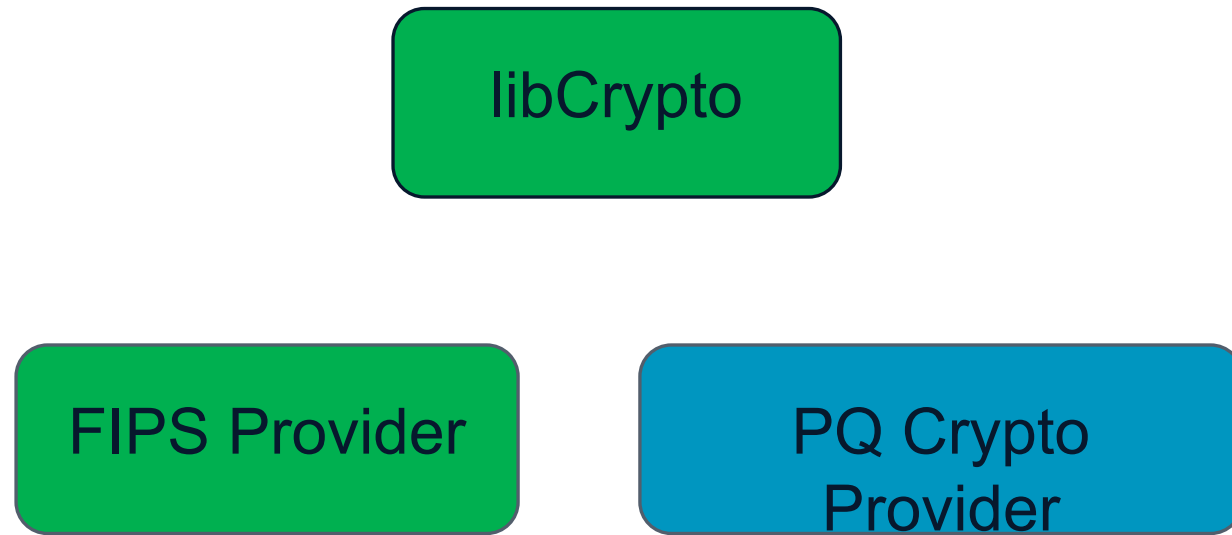
Maximize Shelf-life of Certified Modules

- Hybrid algorithms
- Multiple Providers
- Multiple FIPS Providers

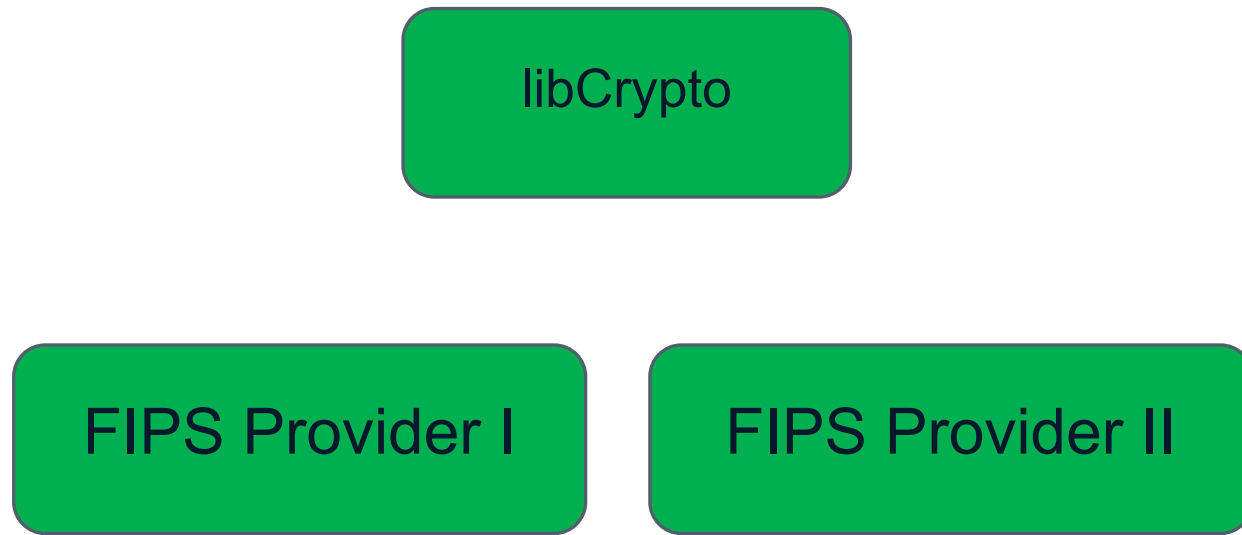
Hybrid Key Example



Crypto Library



Multiple FIPS Providers

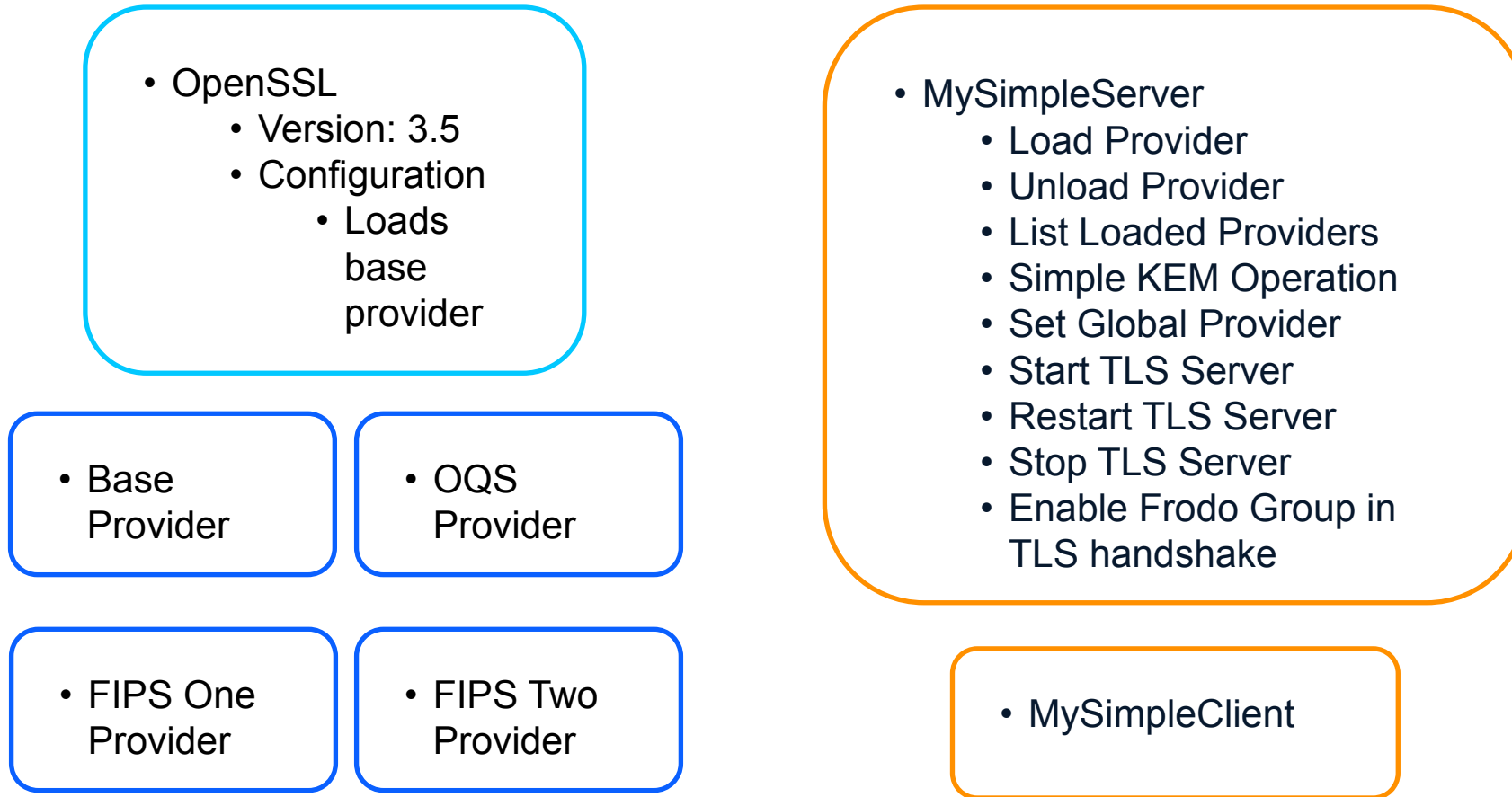


Ryan Hooper

Agenda

- 01 Test Environment
- 02 Default Algorithm Selection
- 03 EVP propquery
- 04 EVP_set_default_properties
- 05 Use Case #1 Swap out Provider
- 06 Use Case #2 Use New Cryptography Algorithms in TLS Server

What does my environment look like?



• Fun Fact: If no provider is loaded via the config OpenSSL will load the Default Provider

Load two Providers that provide the same algorithms

- MySimpleServer
 - listproviders
 - loadprovider fips_one
 - loadprovider fips_two
 - listproviders

```
listproviders
Providers:
Number of providers: 3
  base available: 1
    Name: OpenSSL Base Provider
    Version: 3.5.1
    Build Info: 3.5.1-dev
    Status: 1
  fips_one available: 1
    Name: OpenSSL FIPS Provider
    Version: 3.5.1
    Build Info: 3.5.1-dev
    Status: 1
  fips_two available: 1
    Name: OpenSSL FIPS Provider
    Version: 3.5.1
    Build Info: 3.5.1-dev
    Status: 1
```

Using an algorithm that both Providers provide

- MySimpleServer
 - kemexample MLKEM512

- FIPS One: GEN INIT ML KEM key
- FIPS One: GEN ML KEM key
- FIPS One: Creating ML KEM key
- FIPS One: GEN ML KEM key keysize: 512
- FIPS One: GEN INIT ML KEM key
- FIPS One: GEN ML KEM key
- FIPS One: Creating ML KEM key

```
kemexample MLKEM512
```

```
Testing KEM: MLKEM512
```

```
Testing of KEM split operation: MLKEM512
```

```
SECENC:
```

```
C6 68 A8 F8 48 B6 45 08 12 6D CA 25 07 04 9C CB  
1A 26 EB 15 31 CD BD E2 60 95 C8 34 A1 0B 7E 9B
```

```
SECDEC:
```

```
C6 68 A8 F8 48 B6 45 08 12 6D CA 25 07 04 9C CB  
1A 26 EB 15 31 CD BD E2 60 95 C8 34 A1 0B 7E 9B
```

Load two Providers that provide the same algorithms but swap loading order

- MySimpleServer
 - listproviders
 - loadprovider fips_two
 - loadprovider fips_one
 - listproviders

```
listproviders
Providers:
Number of providers: 3
  base available: 1
    Name: OpenSSL Base Provider
    Version: 3.5.1
    Build Info: 3.5.1-dev
    Status: 1
  fips_one available: 1
    Name: OpenSSL FIPS Provider
    Version: 3.5.1
    Build Info: 3.5.1-dev
    Status: 1
  fips_two available: 1
    Name: OpenSSL FIPS Provider
    Version: 3.5.1
    Build Info: 3.5.1-dev
    Status: 1
```

Once again, using an algorithm that both Providers provide

- MySimpleServer
 - kemexample MLKEM512

- FIPS Two: GEN INIT ML KEM key
- FIPS Two: GEN ML KEM key
- FIPS Two: Creating ML KEM key
- FIPS Two: GEN ML KEM key keysize: 512
- FIPS Two: GEN INIT ML KEM key
- FIPS Two: GEN ML KEM key
- FIPS Two: Creating ML KEM key

```
kemexample MLKEM512
```

```
Testing KEM: MLKEM512
```

```
Testing of KEM split operation: MLKEM512
```

```
SECENC:
```

```
14 3C C1 33 17 1E 57 74 8E B7 F9 EF 3C 89 62 A1  
1C 1C A6 C8 74 D9 F2 64 25 DA E1 CF 71 BB C5 22
```

```
SECDEC:
```

```
14 3C C1 33 17 1E 57 74 8E B7 F9 EF 3C 89 62 A1  
1C 1C A6 C8 74 D9 F2 64 25 DA E1 CF 71 BB C5 22
```

- Fun Fact: It looks like the First Provider loaded that provides the algorithm will be selected*

Starting Point

- MySimpleServer
 - listproviders
 - loadprovider fips_one
 - loadprovider fips_two
 - listproviders

```
listproviders
Providers:
Number of providers: 3
  base available: 1
    Name: OpenSSL Base Provider
    Version: 3.5.1
    Build Info: 3.5.1-dev
    Status: 1
  fips_one available: 1
    Name: OpenSSL FIPS Provider
    Version: 3.5.1
    Build Info: 3.5.1-dev
    Status: 1
  fips_two available: 1
    Name: OpenSSL FIPS Provider
    Version: 3.5.1
    Build Info: 3.5.1-dev
    Status: 1
```

Using EVP's propquery argument

- MySimpleServer
 - kemexample MLKEM512 fips_two

- FIPS Two: GEN INIT ML KEM key
- FIPS Two: GEN ML KEM key
- FIPS Two: Creating ML KEM key
- FIPS Two: GEN ML KEM key keysize: 512
- FIPS Two: GEN INIT ML KEM key
- FIPS Two: GEN ML KEM key
- FIPS Two: Creating ML KEM key

```
kemexample MLKEM512 fips_two
```

```
Testing KEM: MLKEM512
```

```
Properties: provider=fips_two
```

```
Testing of KEM split operation: MLKEM512
```

```
SECENC:
```

```
F7 F8 45 9A 16 86 1C F2 42 4D FC 61 BD 90 C6 03  
97 5C 75 6B 33 DF 13 B3 4D E8 B8 C7 3F 42 B1 46
```

```
SECDEC:
```

```
F7 F8 45 9A 16 86 1C F2 42 4D FC 61 BD 90 C6 03  
97 5C 75 6B 33 DF 13 B3 4D E8 B8 C7 3F 42 B1 46
```

- Fun Fact: When creating an EVP object you can specify a provider to use via the propquery argument

Using EVP_set_default_properties

- MySimpleServer
 - setglobalprovider fips_two
 - kemexample MLKEM512

- FIPS Two: GEN INIT ML KEM key
- FIPS Two: GEN ML KEM key
- FIPS Two: Creating ML KEM key
- FIPS Two: GEN ML KEM key keysize: 512
- FIPS Two: GEN INIT ML KEM key
- FIPS Two: GEN ML KEM key
- FIPS Two: Creating ML KEM key

```
kemexample MLKEM512
```

```
Testing KEM: MLKEM512
```

```
Testing of KEM split operation: MLKEM512
```

```
SECENC:
```

```
16 5B D4 40 4E 5E 46 C3 24 6A DA D5 49 17 EA CD  
2A AA B2 D9 EE FF 7C 67 95 DB 42 31 E9 20 0F 8D
```

```
SECDEC:
```

```
16 5B D4 40 4E 5E 46 C3 24 6A DA D5 49 17 EA CD  
2A AA B2 D9 EE FF 7C 67 95 DB 42 31 E9 20 0F 8D
```

- Fun Fact: You can set a preferred global provider via the EVP_set_default_properties for all algorithms.

Ways to Obtain an Algorithm from Specific Provider

- Using propquerty
- Setting Global Properties

Use Case #1 Swap out Provider

Starting Point

- MySimpleServer
 - listproviders
 - loadprovider fips_one
 - listproviders

```
listproviders
Providers:
Number of providers: 2
  base available: 1
    Name: OpenSSL Base Provider
    Version: 3.5.1
    Build Info: 3.5.1-dev
    Status: 1
  fips_one available: 1
    Name: OpenSSL FIPS Provider
    Version: 3.5.1
    Build Info: 3.5.1-dev
    Status: 1
```

Start TLS Server and have Client make connection

- MySimpleServer
 - TLSServer

- MySimpleClient
 - ./mySimpleClient server server_openssl kem MLKEM512
 - Connect to server_openssl using MLKEM512 as the Group

- FIPS One: GCM INIT
- FIPS One: GCM Update
- FIPS One: GCM Update
- FIPS One: GCM final
- FIPS One: GEN INIT ML KEM key
- FIPS One: GEN ML KEM key
- FIPS One: Creating ML KEM key
- FIPS One: GCM INIT
- FIPS One: GCM Update
- FIPS One: GCM Update
- FIPS One: GCM final
- FIPS One: GCM INIT
- FIPS One: GCM Update
- FIPS One: GCM Update
- FIPS One: GCM final

During the connection update EVP_set_default_properties to FIPS Two

- MySimpleServer
 - loadprovider fips_two
 - setglobalprovider fips_two

- MySimpleClient
 - ./mySimpleClient server server_openssl kem MLKEM512
 - Connect to server_openssl using MLKEM512 as the Group
 - Made **before** the setglobalprovider fips_two call

- FIPS One: GCM INIT
- FIPS One: GCM Update
- FIPS One: GCM Update
- FIPS One: GCM final
- FIPS Two: GEN INIT ML KEM key
- FIPS Two: GEN ML KEM key
- FIPS Two: Creating ML KEM key
- FIPS One: GCM INIT
- FIPS One: GCM Update
- FIPS One: GCM Update
- FIPS One: GCM final
- FIPS One: GCM INIT
- FIPS One: GCM Update
- FIPS One: GCM Update
- FIPS One: GCM final

- Fun Fact: Only new instantiations of an algorithm will use the new globally preferred provider

During the connection Unload FIPS One

- MySimpleServer
 - unloadprovider fips_one
 - listproviders

```
listproviders
Providers:
Number of providers: 2
  base available: 1
    Name: OpenSSL Base Provider
    Version: 3.5.1
    Build Info: 3.5.1-dev
    Status: 1
  fips_two available: 1
    Name: OpenSSL FIPS Provider
    Version: 3.5.1
    Build Info: 3.5.1-dev
    Status: 1
```

- FIPS One: GCM INIT
- FIPS One: GCM Update
- FIPS One: GCM Update
- FIPS One: GCM final
- FIPS Two: GEN INIT ML KEM key
- FIPS Two: GEN ML KEM key
- FIPS Two: Creating ML KEM key
- FIPS One: GCM INIT
- FIPS One: GCM Update
- FIPS One: GCM Update
- FIPS One: GCM final
- FIPS One: GCM INIT
- FIPS One: GCM Update
- FIPS One: GCM Update
- FIPS One: GCM final

- Fun Fact: Even though FIPS One has been unloaded its algorithms are still being used by the SSL_CTX and the SSL Server.

What if a second connection is made after FIPS One has been unloaded

- MySimpleClient
 - ./mySimpleClient server server_openssl kem MLKEM512
 - Connect to server_openssl using MLKEM512 as the Group
 - Made **after** FIPS One has been unloaded

- FIPS One: GCM INIT
- FIPS One: GCM Update
- FIPS One: GCM Update
- FIPS One: GCM final
- FIPS Two: GEN INIT ML KEM key
- FIPS Two: GEN ML KEM key
- FIPS Two: Creating ML KEM key
- FIPS One: GCM INIT
- FIPS One: GCM Update
- FIPS One: GCM Update
- FIPS One: GCM final
- FIPS One: GCM INIT
- FIPS One: GCM Update
- FIPS One: GCM Update
- FIPS One: GCM final

- Fun Fact: Even for new SSL Client Connections the Server will continue to use FIPS One for TLS and FIPS Two for the Application

What happens if we Restart the TLS Server

- MySimpleServer
 - RestartTLSServer

- MySimpleClient
 - ./mySimpleClient server server_openssl kem MLKEM512
 - Connect to server_openssl using MLKEM512 as the Group
 - Both Connections start to use FIPS Two

```
• FIPS Two: GCM INIT
• FIPS Two: GCM Update
• FIPS Two: GCM Update
• FIPS Two: GCM final
• FIPS Two: GEN INIT ML KEM key
• FIPS Two: GEN ML KEM key
• FIPS Two: Creating ML KEM key
• FIPS Two: GCM INIT
• FIPS Two: GCM Update
• FIPS Two: GCM Update
• FIPS Two: GCM final
• FIPS Two: GCM INIT
• FIPS Two: GCM Update
• FIPS Two: GCM Update
• FIPS Two: GCM final
```

- Fun Fact: Restarting the TLS Server recreates the SSL_CTX and now the TLS connection is using FIPS Two

Use Case #2 Use New Cryptography Algorithms in TLS Server

Updated Starting Point

- MySimpleServer
 - listproviders
 - loadprovider fips_one
 - loadprovider fips_two
 - loadprovider oqsprovider
 - listproviders

```
listproviders
Providers:
Number of providers: 4
  base available: 1
    Name: OpenSSL Base Provider
    Version: 3.5.1
    Build Info: 3.5.1-dev
    Status: 1
  fips_one available: 1
    Name: OpenSSL FIPS Provider
    Version: 3.5.1
    Build Info: 3.5.1-dev
    Status: 1
  fips_two available: 1
    Name: OpenSSL FIPS Provider
    Version: 3.5.1
    Build Info: 3.5.1-dev
    Status: 1
  oqsprovider available: 1
    Name: Cisco_PQC_Provider3
    Version: 1.0.0
    Build Info: OSS_CSM_PQC-PROVIDER_0.8.1 CSCO_CSM_CISCO-PQC-P
ROVIDER_1.0.0 based on Cisco_PQC_Library v.OSS_CSM_LIBOQS_0.13.0 CS
CO_CSM_CISCO-PCQ-LIB_1.0.0
    Status: 1
```

Start TLS Server and have Client attempt Connection with frodo640aes

- MySimpleServer
 - TLSServer

- MySimpleClient
 - `./mySimpleClient server server_openssl kem frodo640aes`
 - Connect to `server_openssl` using `frodo640aes` as the Group

```
root@45b829a7d13b:/app# ./mySimpleClient server server_openssl kem frodo640aes
KEM algorithm set to: frodo640aes
Server_port: server_openssl:7176
Setting KEM Algorithm to: frodo640aes
Failed to connect to server
Error in BIO_do_connect:
  error:0A000410:SSL routines::ssl/tls alert handshake failure
```

- Fun Fact: Just loading the provider is not enough

Start TLS Server with an updated group list that includes frodo640aes and have Client attempt Connection with frodo640aes

- MySimpleServer
 - EnableFrodoGroup
 - TLSServer/RestartTLSServer

- MySimpleClient
 - ./mySimpleClient server server_openssl kem frodo640aes
 - Connect to server_openssl using frodo640aes as the Group
 - Connection Now succeeds

```
root@45b829a7d13b:/app# ./mySimpleClient server server_openssl kem frodo640aes
KEM algorithm set to: frodo640aes
Server_port: server_openssl:7176
Setting KEM Algorithm to: frodo640aes
SSL connection established successfully
```

- Fun Fact: To be dynamic you must update your TLS configuration to reflect the changes to your cryptography.

What if I tried updating my group list via SSL_CTX_set1_groups_list without uploading the OQS Provider that provides frodo640aes

- MySimpleServer
 - EnableFrodoGroup

```
Cert File: cert.pem Key File:: key.pem  
Failed to set FRODO algorithm
```

- MySimpleClient
 - ./mySimpleClient server server_openssl
kem frodo640aes
 - Connect to server_openssl using
frodo640aes as the Group
 - Connection fails

- Fun Fact: The call to SSL_CTX_set1_groups_list will error out if it cannot find all the algorithms specified in the list