presented by

Microsoft

Post Quantum Computing: What You Need To Know About the UEFI Forum Updates

UEFI 2026 Webinars

April 30, 2026

Jiewen Yao (Intel) and Sean Brogan (Microsoft)

Meet the Presenters

Jiewen Yao

Principal Engineer, Intel

Jiewen Yao has been engaged as a firmware developer for more than 20 years. He is chair of UEFI Security Sub Team and co-chair of TCG PC Client Working Group. He is EDK2 maintainer of SecurityPkg and CryptoPkg.

Meet the Presenters

Sean Brogan

SW Engineering Manager, Microsoft

Sean has been a firmware developer for more than 20 years. He manages the Microsoft Core UEFI team, represents Microsoft in the UEFI Forum, and is actively involved in numerous related open-source projects (Project Mu, ODP Patina, and Edk2).

Disclaimer

- The content in this presentation reflects a point-in-time snapshot of the work conducted by the UEFI Security Sub-Team (USST) and is subject to change without notice
- The final published UEFI Specification shall serve as the sole normative reference
- This content is provided for informational purposes only and should not be used as the basis for any production implementation

Agenda

- Introduction
- UEFI Spec PQC Plan
- Windows PQC Plan
- Summary

Introduction

- Post-quantum cryptography (PQC) algorithm
  - FIPS 203 (ML-KEM)
  - FIPS 204 (ML-DSA)
  - FIPS 205 (SLH-DSA)

- PQC requirement
  - US NSA, EU, UK NCSC, Germany BSI, France ANSSI, …

- UEFI Secure Boot touch point
  - How UEFI binaries are authenticated (code signing verification)
  - How execution is authorized using the allow list (db)
  - How execution is revoked or blocked using the revoke list (dbx)
  - How updates to PK, KEK, db, and dbx are authenticated and authorized

Goals

- Add PQC Support
- Address 2011 Key rolling challenges
- Adjust features/capabilities based on over a decade of usage

Principles

- Agility and infrastructure: help slow moving firmware survive in the quickly evolving threat landscape
- Stay away from specifics: Security requirements vary by geography and regulation
- Expect change: PQC maturity is low

Out of Scope

- Mandate any specific PQC algorithm
- Define new PQC crypto primitives or hybrid combinations
- Define measurement requirement (deferred to TCG)

UEFI Spec PQC Update Plan

- **A. Add New Feature**
  - 1. Allow Multiple Signature verification (fix DBX policy)
  - 2. Mandate the order of Multiple Signature verification
  - 3. Add TBSCertificate (EFI_CERT_X509_SHAxxx_GUID) for DB
  - 4. Mandate TBSCertificate or HASH for DBX and Remove Certificate(EFI_CERT_X509_GUID) usage from DBX
  - 5. Add EFI Crypto Indicator Table (ECIT)
  - 6. Add check for supported algorithm when PK/KEK/db/dbx in enrolled
  - 7. Mandate one SignatureType only for SetVariable() call
  - 8. Add KEK self-signed append operations

- **B. Deprecate/Remove Old Feature**
  - 1. Deprecate "SignatureSupport" variable (replaced by ECIT)
  - 2. Deprecate Private AuthVariable
  - 3. Remove EFI_VARIABLE_AUTHENTICATION_3 descriptor
  - 4. Remove Audit Mode and Deployed Mode from Secure boot

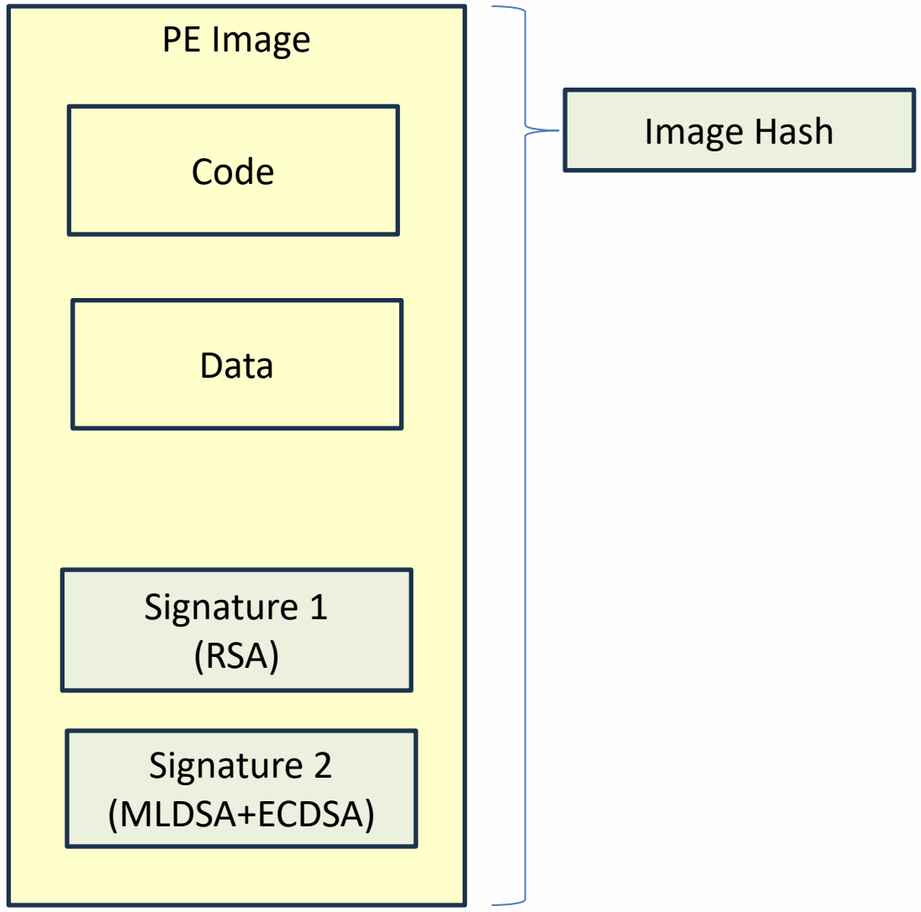
- **C. Clarification/Clean up**
  - 1. Clarify to ignore cert validity check in secure boot
  - 2. Clarify all hash algorithms (EFI_SHAxxx_GUID) for DBX
  - 3. Clarify to deprecate SHA1
  - 4. Clean up the language that limits Crypto Agility
  - 5. Clarify the number of SignerInfo in AuthVariable update

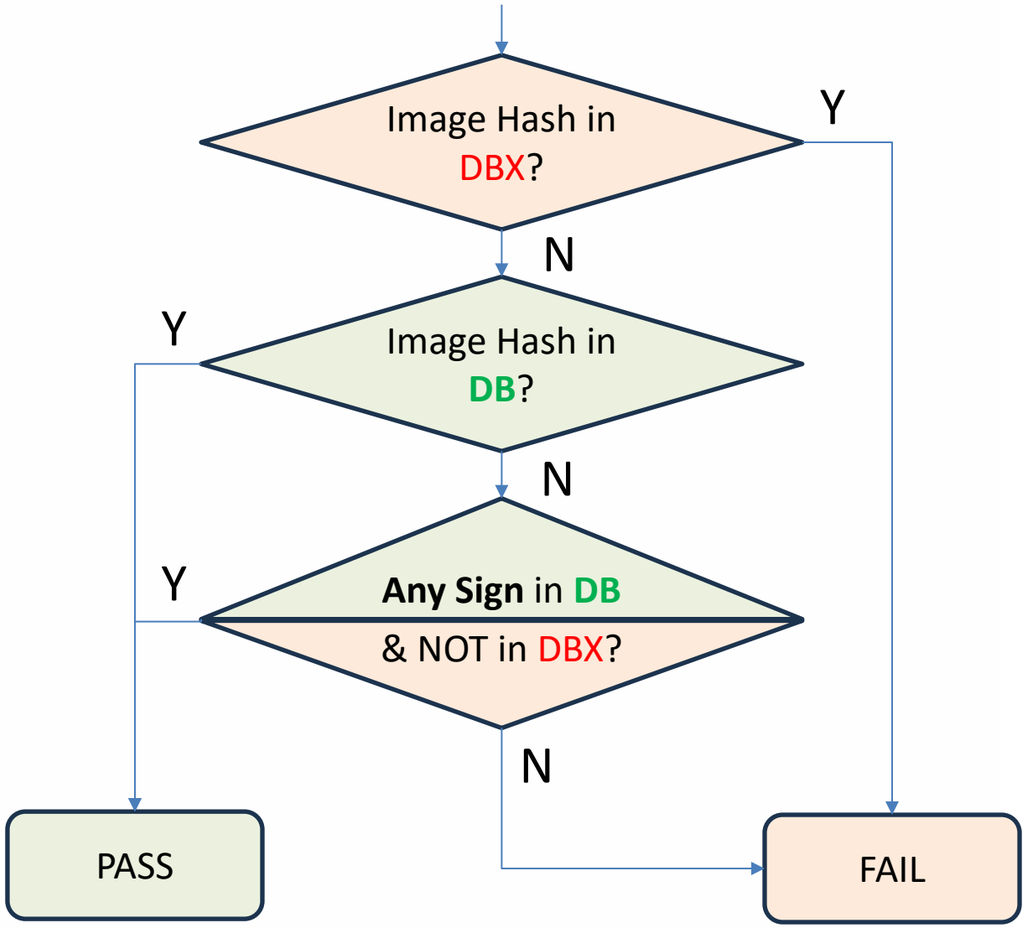
A1/A2. Multiple Signature Verification

- Problem Statement
  - During PQC transition time, we expect a UEFI image may need to support multiple signatures (crypto agile)
  - As long as **one of** multiple signatures is validated, then the image **passes** the verification
  - But current spec says: "**neither** the hash of the binary **nor any** present signature is reflected in **dbx**", it blocks the expected multiple signature usage

- Proposal
  - Clarify the dbx policy. Verification order:
    - If the hash of image in DBX – FAIL
    - Else if the hash of image in DB – PASS
    - Else if one of image signature is in DB but not in DBX – PASS
    - Else – FAIL
  - Mandate the order of multiple signature verification
    - Top down based on signature in UEFI image

A1/A2. Multiple Signature Verification



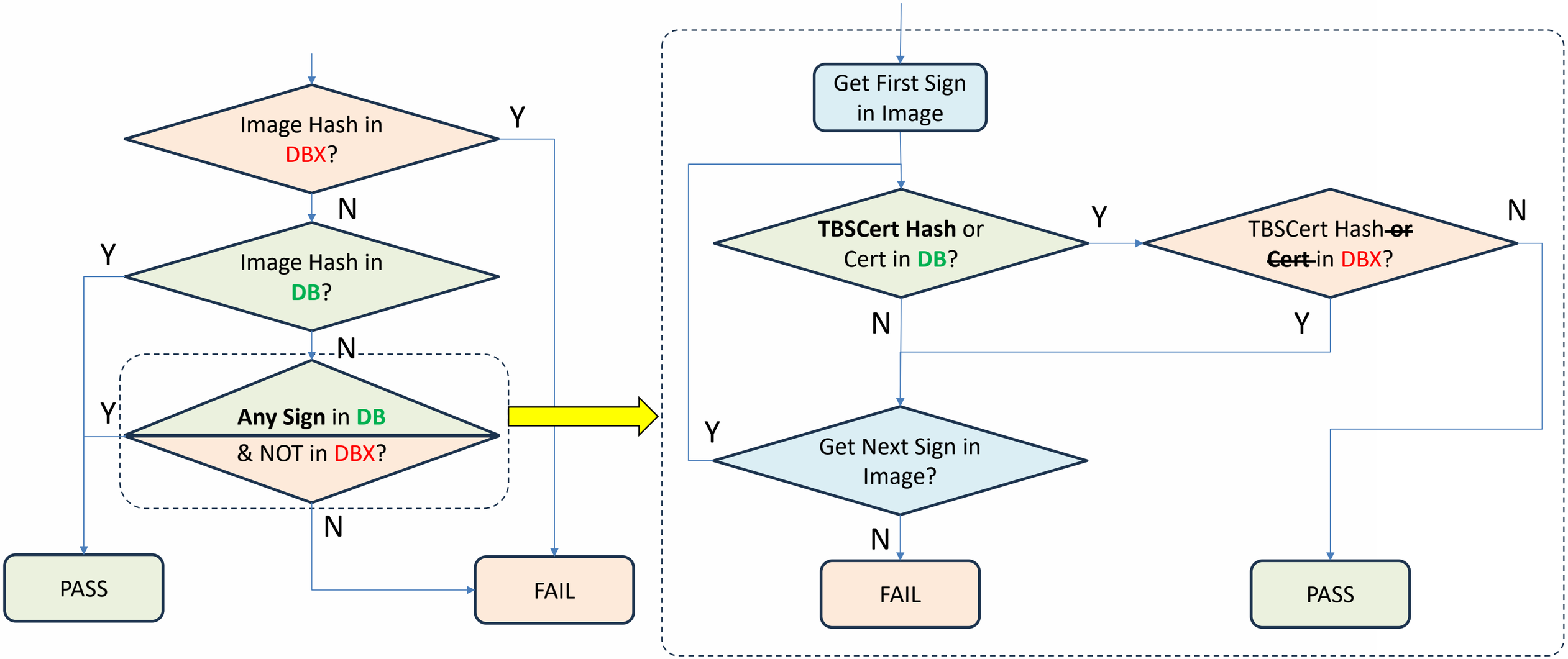


A3/A4. TBSCertificate in DB, No Cert in DBX

- Problem Statement
  - Putting a full PQC cert in DB may consume lots of flash space
  - Putting a full PQC cert in DBX may consume lots of flash space, and cause problem if we want to revoke multiple PQC cert at one time (because we will do algo check)

- Proposal
  - Add TBSCertificate Hash for DB
  - Remove Certificate from DBX
  - The final database may include:
    - 1) (DB/DBX) Hash of Image (EFI_CERT_SHA256/SHA384/SHA512_GUID)
    - 2) (**DB**/DBX) TBSCertificate Hash (EFI_CERT_X509_SHA256/SHA384/SHA512_GUID)
    - 3) (DB~~/DBX~~) Full Certificate (EFI_CERT_X509_GUID)

A3/A4. TBSCertificate in DB, No Cert in DBX



A5. EFI Crypto Indicator Table

- Problem Statement
  - An OS loader/application may want to know if the UEFI firmware is PQC ready
  - But there is no existing mechanism to report which crypto algorithm is supported

- Proposal
  - Report an EFI Crypto Indicator Table, as UEFI Configuration Table and ACPI table
  - Scope: The crypto used in the UEFI firmware
  - Extension: Allow OEM/ODM/IBV/... to report OEM/IBV specific usage

A5. EFI Crypto Indicator Table

EFI Crypto Indicator Table

UEFI Image Verification (DB - PKCS7)

1.2.840.113549.1.1.11	(sha256WithRSAEncryption)
1.2.840.113549.1.1.12	(sha384WithRSAEncryption)
1.2.840.113549.1.1.13	(sha512WithRSAEncryption)
1.2.840.10045.4.3.2	(ecdsa-with-SHA256)
1.2.840.10045.4.3.3	(ecdsa-with-SHA384)
1.2.840.10045.4.3.4	(ecdsa-with-SHA512)
2.16.840.1.101.3.4.3.17	(id-ml-dsa-44)
2.16.840.1.101.3.4.3.18	(id-ml-dsa-65)
2.16.840.1.101.3.4.3.19	(id-ml-dsa-87)

UEFI Authenticated Variable (PK/KEK - PKCS7)

1.2.840.113549.1.1.11	(sha256WithRSAEncryption)
1.2.840.113549.1.1.12	(sha384WithRSAEncryption)
1.2.840.113549.1.1.13	(sha512WithRSAEncryption)
1.2.840.10045.4.3.2	(ecdsa-with-SHA256)
1.2.840.10045.4.3.3	(ecdsa-with-SHA384)
1.2.840.10045.4.3.4	(ecdsa-with-SHA512)
2.16.840.1.101.3.4.3.17	(id-ml-dsa-44)
2.16.840.1.101.3.4.3.18	(id-ml-dsa-65)
2.16.840.1.101.3.4.3.19	(id-ml-dsa-87)

UEFI Secure Boot Authorization (DB)

A5C059A1-94E4-4AA7-87B5-AB155C2BF072	EFI_CERT_X509
C1C41626-504C-4092-ACA9-41F936934328	EFI_CERT_SHA256
FF3E5307-9FD0-48C9-85F1-8AD56C701E01	EFI_CERT_SHA384
093E0FAE-A6C4-4F50-9F1B-D41E2B89C19A	EFI_CERT_SHA512
3BD2A492-96C0-4079-B420-FCF98EF103ED	EFI_CERT_X509_SHA256
7076876E-80C2-4EE6-AAD2-28B349A6865B	EFI_CERT_X509_SHA384
446DBF63-2502-4CDA-BCFA-2465D2B0FE9D	EFI_CERT_X509_SHA512

UEFI Image Revocation (DBX)

~~A5C059A1-94E4-4AA7-87B5-AB155C2BF072~~	~~EFI_CERT_X509~~
C1C41626-504C-4092-ACA9-41F936934328	EFI_CERT_SHA256
FF3E5307-9FD0-48C9-85F1-8AD56C701E01	EFI_CERT_SHA384
093E0FAE-A6C4-4F50-9F1B-D41E2B89C19A	EFI_CERT_SHA512
3BD2A492-96C0-4079-B420-FCF98EF103ED	EFI_CERT_X509_SHA256
7076876E-80C2-4EE6-AAD2-28B349A6865B	EFI_CERT_X509_SHA384
446DBF63-2502-4CDA-BCFA-2465D2B0FE9D	EFI_CERT_X509_SHA512

UEFI Secure Boot Servicing Authorization (KEK/PK)

A5C059A1-94E4-4AA7-87B5-AB155C2BF072	EFI_CERT_X509

UEFI Device Firmware Update (ESRT)

. . .

UEFI System Firmware Update (FMP)

. . .

Extension: Intel Boot Guard

. . .

Extension: OEM

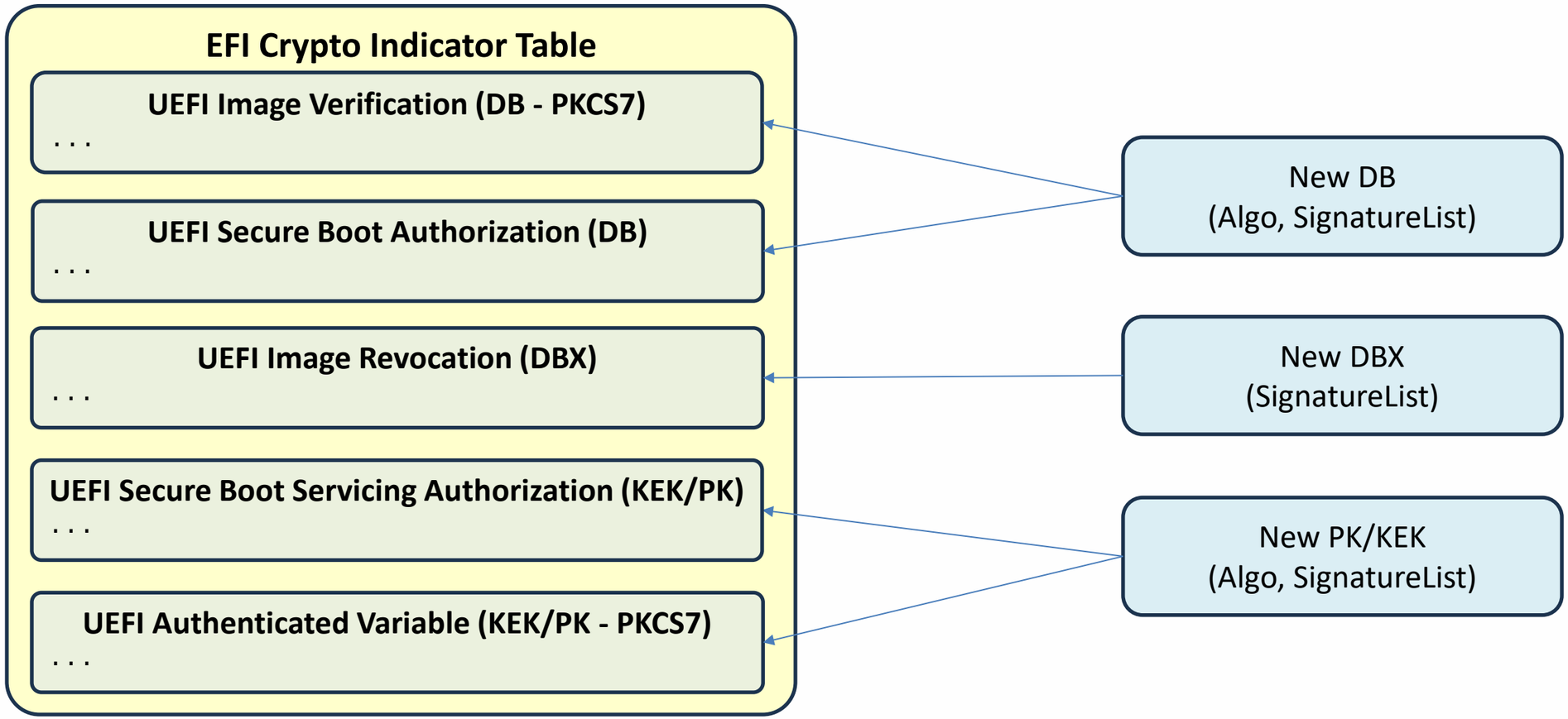
. . .

A6. Algo Check for PK/KEK/db/dbx Enroll

- Problem Statement
  - Currently, there is no algo check for PK/KEK/db/dbx, when it is updated
  - If a crypto algo is unsupported by the UEFI, it still returns SUCCESS. But it will cause image verification failure later

- Proposal
  - Add text clearly, that UEFI needs to check and reject if the updated PK/KEK/db/dbx includes unsupported algorithm
  - (The caller can query ECIT to understand which algo is supported)

A6. Algo Check for PK/KEK/db/dbx Enroll



A7. Mandate One EFI_SIGNATURE_LIST Only for SetVar

- Problem Statement
  - Since there is check for PK/KEK/db/dbx, the caller need to know which one is not supported if the SetVar returns UNSUPPORTED

- Proposal
  - Mandate SetVar only include one EFI_SIGNATURE_LIST
  - If it is rejected, then the caller can know clearly why it is rejected

A7. Mandate One EFI_SIGNATURE_LIST Only for SetVar

EFI_VARIABLE_AUTHENTICATION_2 descriptor

TimeStamp

AuthInfo (WIN_CERTIFICATE_UEFI_GUID)

Hdr (WIN_CERTIFICATE)

```
dwLength
wRevision = 0x0200
wcertificateType = WIN_CERT_TYPE_EFI_GUID (0x0EF1)
```

CertType = EFI_CERT_TYPE_PKCS7_GUID

CertData = PKCS7 SignedData

Var Data

EFI_SIGNATURE_LIST(0) (Only One)

```
SignatureType - EFI_CERT_SHA256|384|512_GUID |
                EFI_CERT_X509_GUID |
                EFI_CERT_X509_SHA256|384|512_GUID
```

EFI_SIGNATURE_DATA (0)

…

EFI_SIGNATURE_DATA (m)

Image Database Variable

EFI_SIGNATURE_LIST(0)

```
SignatureType (EFI_CERT_SHA384_GUID)
SignatureListSize
```

EFI_SIGNATURE_DATA(0)

```
SignatureOwner (GUID)
SignatureData
```

…

EFI_SIGNATURE_DATA(m)

```
SignatureOwner (GUID)
SignatureData
```

…

EFI_SIGNATURE_LIST(n)

```
SignatureType (EFI_CERT_X509_SHA384_GUID)
SignatureListSize
```

EFI_SIGNATURE_DATA(0)

```
SignatureOwner (GUID)
SignatureData
```

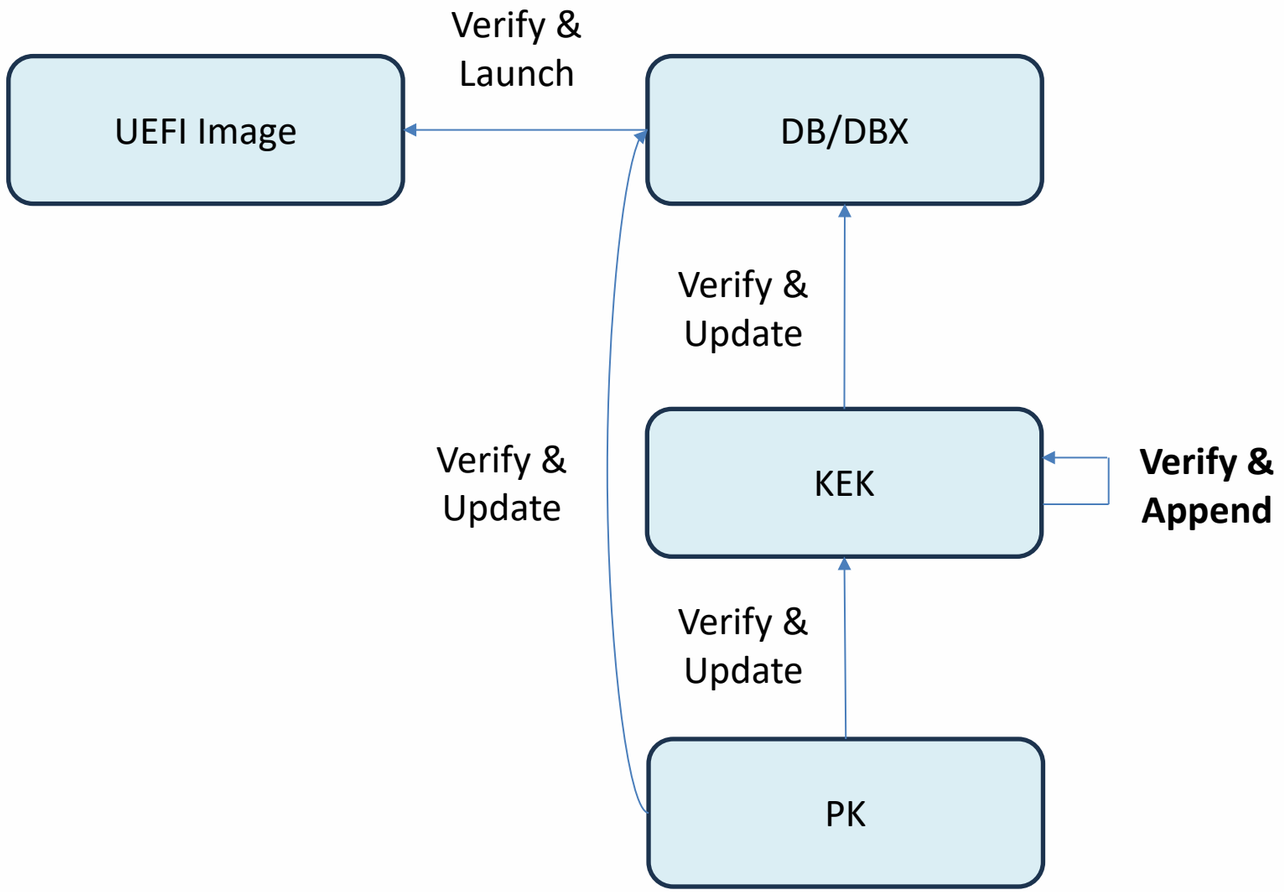
…

A8. KEK Self-Signed Append

- Problem Statement
  - Currently, KEK can only be updated if it is signed by PK
  - It may cause delay, if OS issues a new KEK

- Proposal
  - Allow a new KEK to be appended if it is signed by old KEK
  - That allows the OSV can issue and append KEK for update

A8. KEK Self-Signed Append



Windows OS Usage

- Windows teams involved in USST discussions and nearly all proposals
- Windows will use WHCP and existing communication channels to share additional updates
  - Algorithm specifics still pending final alignment
  - Min UEFI version likely 2.12 for PQC platforms
  - ECIT will be required

OS Secure Boot Enrollment

- Updated manufacturing guidance encouraging shipping in “Setup Mode”
  - Attack Surface Reduction / Servicing Reduction
  - User Intent
  - Platform agility over time
- Potential new capabilities (pending)
  - Enroll during Windows OOBE (setup)
  - Enroll from Windows OS
  - View enrolled keys in Windows OS

Secure Boot Servicing – DBX Updates

- Advancements necessary to deal with more diverse device ecosystem and capabilities
  - Existing pre UEFI 2.12
  - ECIT and algorithm
  - Multi-signing boot manager
- Expect continued use of SHA256 revocations

UEFI CA PQC Plans

- For option-roms, shims, and UEFI applications
- Follow current behavior with 2011/2023 UEFI CA
- Multi-sign with various active keys

Summary

- Many critical changes are coming
- Join the code first effort to prepare your platforms and help shape the future
  - For more information on the UEFI PQC efforts, please view the white paper “Post-Quantum Cryptography: UEFI Specification Updates” on the UEFI Forum website: https://uefi.org/learning_center/papers
  - Edk2 Project tracking: Backlog · PQC Code First

Acknowledgement (USST Members)

- Alex Podgorsky (AMI)
- Andrei Popov (Microsoft)
- Bill Munger (Dell)
- Chow Jeremy (Dell)
- Collin Parker (Lenovo)
- Doug Flick (Microsoft)
- Goutam Hegde (Cisco)
- Inbal Levi (Microsoft)
- James Bottomley (Linux)
- Jason Fisher (Microsoft)
- Jordan Geurten (Microsoft)
- Jordan Rogers (Microsoft)
- Kevin Davis (Insyde)
- Lenny Szubowicz (Red Hat)
- Michael Kinney (Intel)
- Mike Demeter (Lenovo)
- Nikolay Kalaichidi (Dell)
- Patrick Gibbons (Lenovo)
- Peter Jones (Red Hat)
- Prasanna Karthik Mutharaju (Microsoft)
- Rick Bramley (HP Inc)
- Sachin Ganesh (AMI)
- Scott Shell (Microsoft)
- Sochi Ogbuanya (Microsoft)
- Srini Narayana (AMI)
- Stuart Yoder (ARM)
- Terry Lee (HP Enterprise)
- Tim Hoppen (Phoenix)
- Tonry Richard (Dell)
- Yi Li (Intel)

Questions?

References – Standard/Guide/…

- NIST
  - FIPS 203, MLKEM: https://csrc.nist.gov/pubs/fips/203/final
  - FIPS 204, MLDSA: https://csrc.nist.gov/pubs/fips/204/final
  - FIPS 205, SLHDSA: https://csrc.nist.gov/pubs/fips/205/final
  - NIST SP 800-227, KEM Recommendation: https://csrc.nist.gov/pubs/sp/800/227/final
  - NIST IR 8547, PQC Transition (draft): https://csrc.nist.gov/pubs/ir/8547/ipd
  - PQC FAQ: https://csrc.nist.gov/projects/post-quantum-cryptography/faqs
- NSA
  - CNSA2.0 Algo: https://media.defense.gov/2025/May/30/2003728741/-1/-1/0/CSA_CNSA_2.0_ALGORITHMS.PDF
  - CNSA2.0 FAQ: https://media.defense.gov/2022/Sep/07/2003071836/-1/-1/0/CSI_CNSA_2.0_FAQ_.PDF
- IETF
  - MLDSA Cert: https://datatracker.ietf.org/doc/rfc9881/
  - MLDSA CMS: https://datatracker.ietf.org/doc/rfc9882/
  - SLHDSA Cert: https://datatracker.ietf.org/doc/rfc9909/
  - SLHDSA CMS: https://datatracker.ietf.org/doc/rfc9814/
  - Composite MLDSA Sig: https://datatracker.ietf.org/doc/draft-ietf-lamps-pq-composite-sigs/
  - Hybrid TLS: https://datatracker.ietf.org/doc/draft-ietf-tls-hybrid-design
- Other
  - UEFI PQC Impact - https://uefi.org/sites/default/files/resources/Post%20Quantum%20Webinar.pdf
  - PQShield: https://pqshield.com/secure-boot-considerations-with-pqc/
  - EU: https://digital-strategy.ec.europa.eu/en/library/coordinated-implementation-roadmap-transition-post-quantum-cryptography
  - UK NCSC: https://www.ncsc.gov.uk/guidance/pqc-migration-timelines
  - UK NCSC: https://www.ncsc.gov.uk/whitepaper/next-steps-preparing-for-post-quantum-cryptography
  - Germany BSI: https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Crypto/PQC-joint-statement.pdf?
  - France ANSSI: https://messervices.cyber.gouv.fr/guides/en-follow-position-paper-post-quantum-cryptography

References – EDKII/Prototype

- EDKII UEFI POC prototype
  - https://github.com/tianocore/edk2-staging/tree/UEFI_PQC

- OpenSSL 4.0 prototype:
  - https://github.com/tianocore/edk2-staging/tree/OpenSSL40

Thanks for attending a UEFI Forum sponsored webinar

For more information on UEFI Forum and UEFI Specifications, visit http://www.uefi.org

presented by