Post-Quantum

Cryptography Conference

Advancing Cryptographic Transparency: CBOM Standardization in CycloneDX

Basil Hess
Senior Research Engineer at IBM Research

KEYFACTOR CRYPTO4A SSL.com ENTRUST HID

October 28 - 30, 2025 - Kuala Lumpur, Malaysia

PKI Consortium

Advancing Cryptographic Transparency: CBOM Standardization in CycloneDX

—

Basil Hess
IBM Research Europe - Zurich

PKI Consortium / PQC Conference
Kuala Lumpur, Malaysia

October 30, 2025

IBM **Research** Security

Agenda & Overview

A Cryptography Bill of Materials (CBOM) is an object model to describe cryptographic assets and their dependencies.

1. Why Cryptography Bill of Materials (CBOM) now?
2. Core challenges in creating actionable CBOMs
3. One specification for all BOMs: CycloneDX
4. The xBOM playbook: CBOM + SBOM + OBOM + HBOM + SaaSBOM + MBOM
5. Tooling & Ecosystem
6. What’s next

Why CBOM?

- Cryptography is everywhere: code, configuration, certificates, services, hardware
- Comprehensive inventory of cryptographic assets is required
  - OMB M-23-02

    > *… software or hardware implementation of one or more cryptographic algorithms that provide one or more of the following services: (1) creation and exchange of encryption keys; (2) encrypted connections; or (3) creation and validation of digital signatures.*

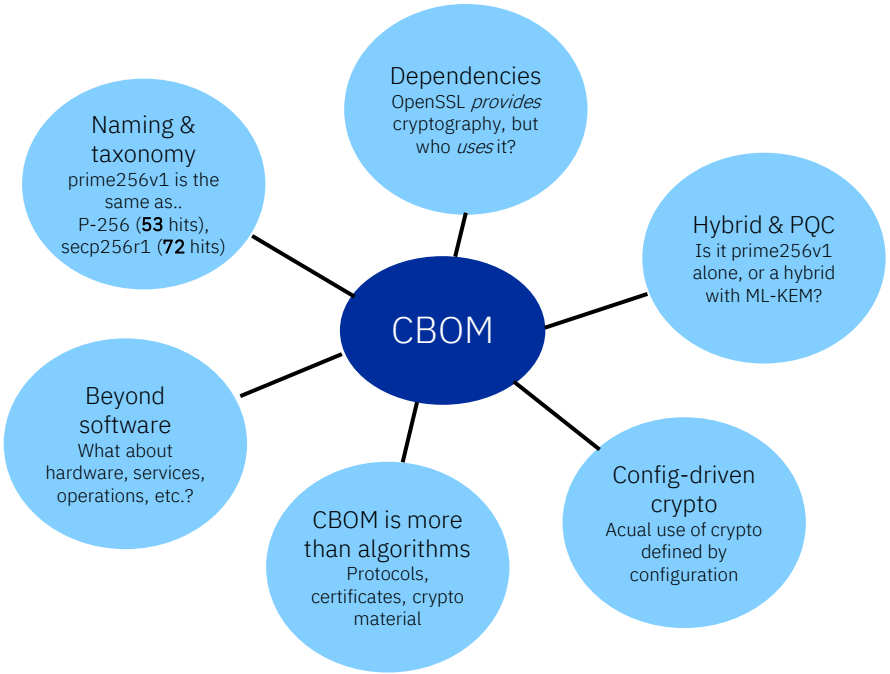
  - EU Roadmap for the Transition to Post-Quantum Cryptography

    > *Member States should promote and support that useful cryptographic inventories are being created and maintained… Using a* ***standardised format*** *for a cryptographic inventory, like CBOM (Cryptographic Bill of Materials, an extension of the SBOM standard), is recommended.*

- CNSA 2.0 sets aggressive PQC migration deadlines (ongoing, full PQC migration by 2033)
- Interoperability matters: we need a standardized CBOM format, enabling interchangeability, automation and trust across vendors and consumers

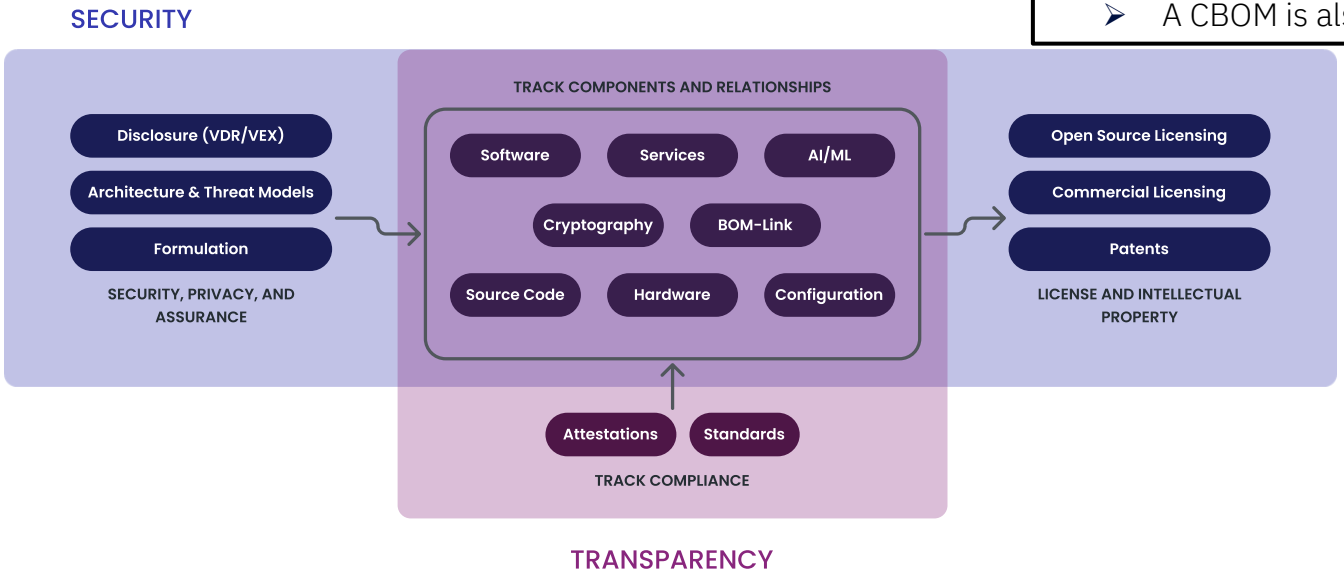
Challenges in creating CBOM

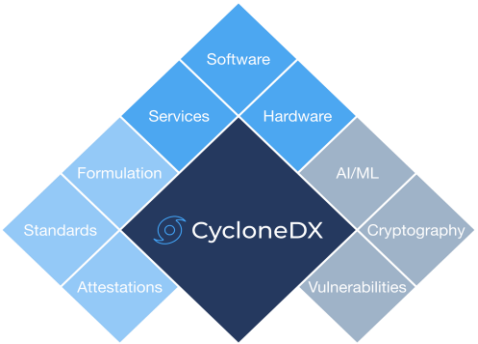
- A grep for primev256v1 in OpenSSL 3.5.3 source code finds **164** occurrences. Is OpenSSL quantum safe? ECC is not quantum-safe.



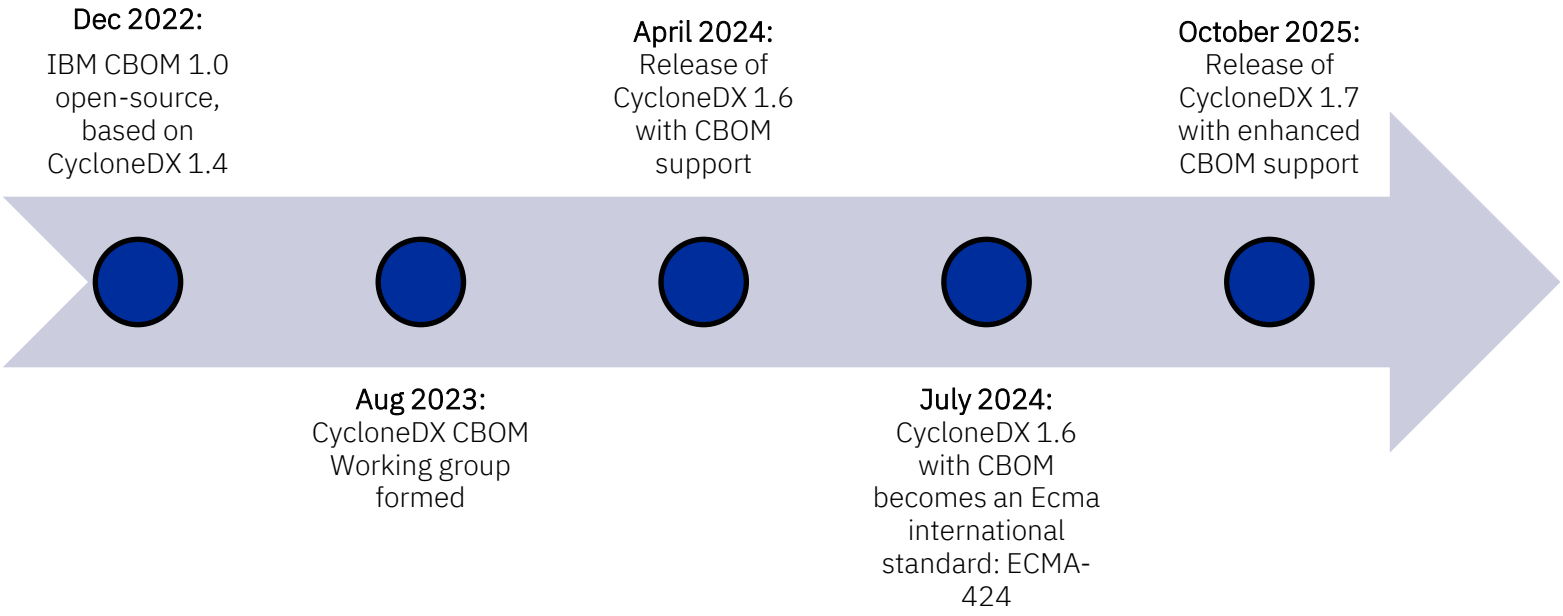
CycloneDX – One spec for all BOMs

- A standard for the software supply chain
- OWASP CycloneDX is a full-stack BOM standard: ECMA-424
- Initially designed for Software, now spans many more Bill of Material use cases.
- A single specification for all xBOM use cases
  - ➢ A CBOM is also an xBOM

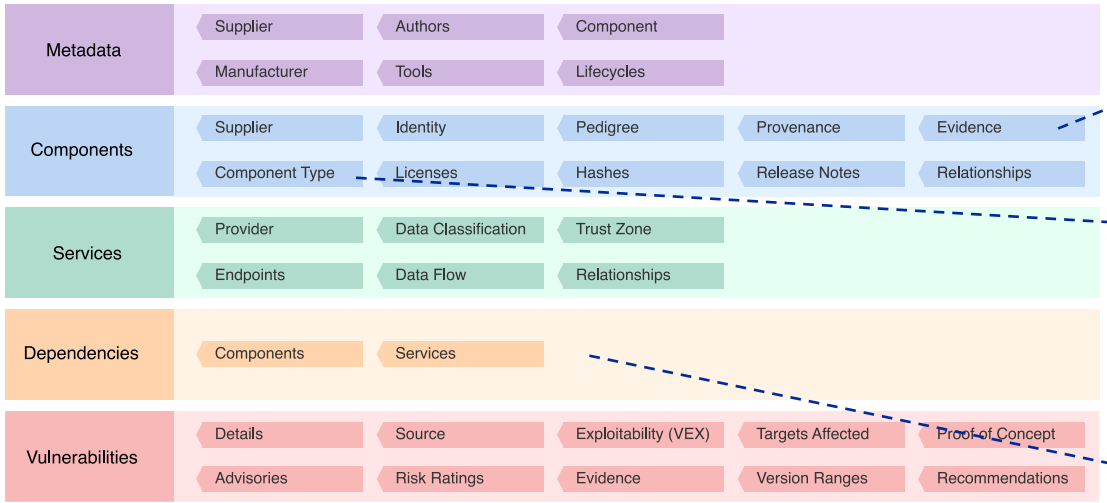




CycloneDX CBOM timeline



Anatomy of a CycloneDX (C)BOM



CBOM extensions to the CycloneDX object model:

occurrence (e.g., location, line, offset, symbol, additionalContext) and *confidence*

cryptographic-asset (algorithm, protocol, certificate, related-crypto-material)
Other component types: application, framework, library, container, device, firmware, ML model, data

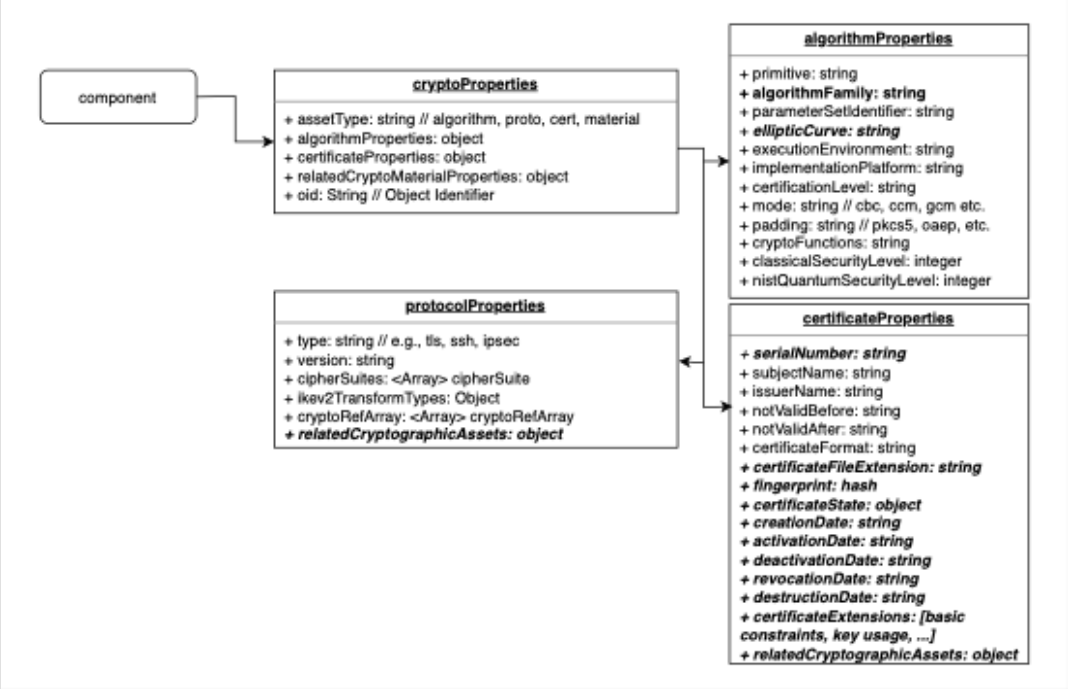
Directed dependencies: *dependsOn* and *provides*

Anatomy of CycloneDX CBOM: Schema

A cryptographic asset is a CycloneDX *component.*

Sub-types are:

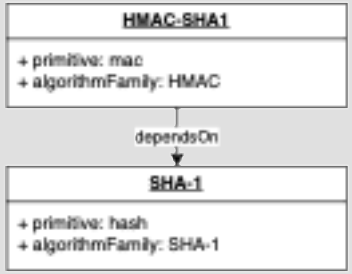
- Algorithms
- Protocols
- Certificates
- Related cryptographic material (e.g., keys, tokens)



Crypto Dependencies: Constructions

SHA-1 is broken…

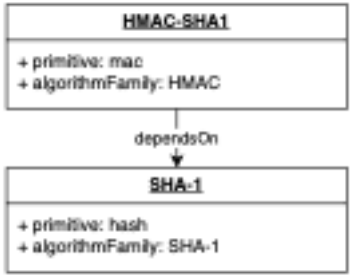
- HMAC-SHA-1
- Self-signed root certificates



Crypto Dependencies: Hybrid PQC

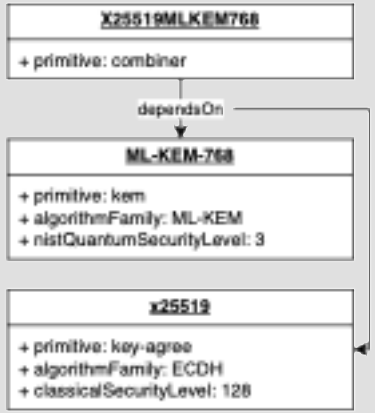
SHA-1 is broken…

- HMAC-SHA-1
- Self-signed root certificates



ECC is not quantum safe…

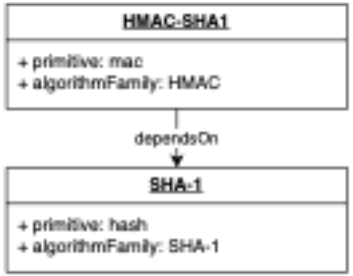
- Hybrids / combiners
- Using ECC + PQC



Crypto Dependencies: Applications and Libraries

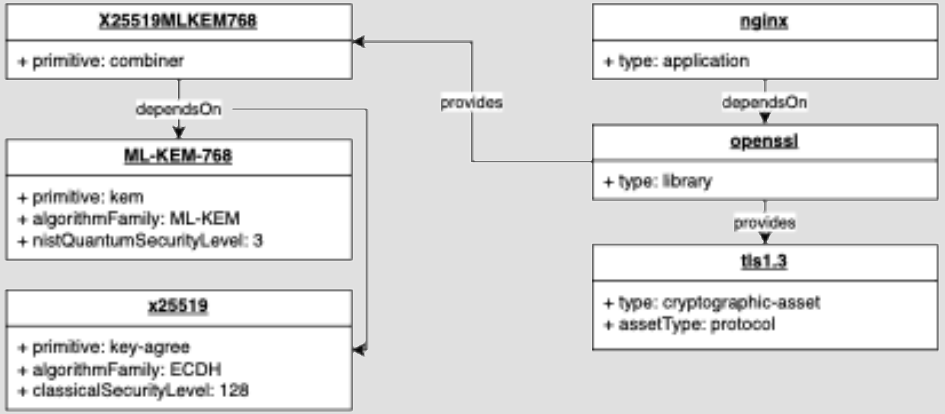
SHA-1 is broken…

- HMAC-SHA-1
- Self-signed root certificates



ECC is not quantum safe…

- Hybrids / combiners
- Using ECC + PQC

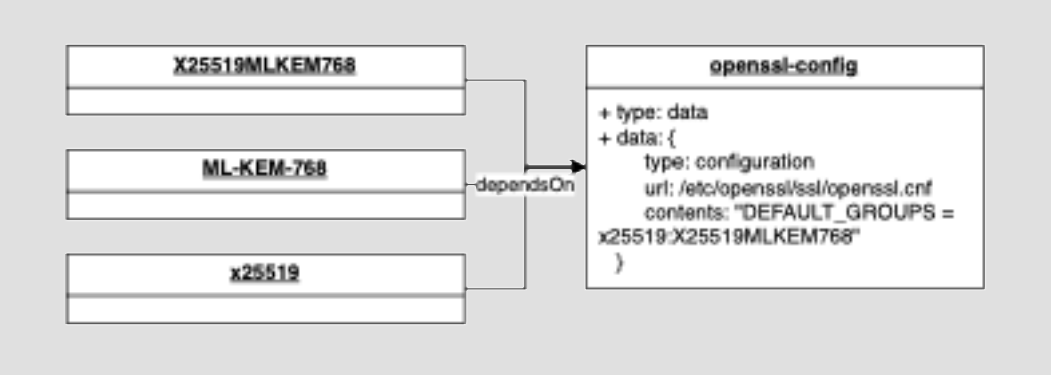


Cryptography is provided by

libraries, used by applications or services

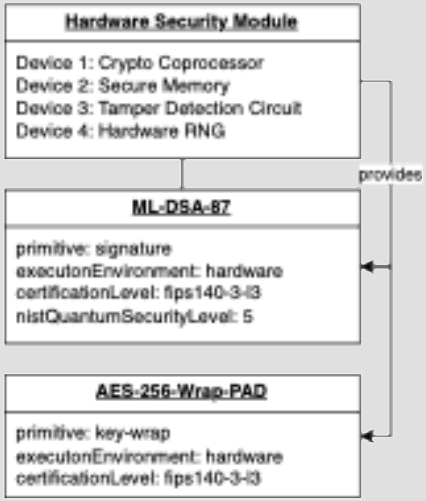
Config-driven Crypto: CBOM + OBOM

- Cipher suites often in configs not code
- Operations Bill of Materials (OBOMs) captures runtime configs and links to CBOM
- Examples: OpenSSL config file enabling hybrid (PQC/classical) KEM



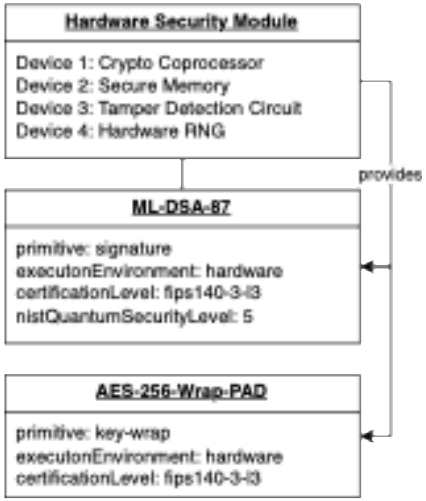

```
"components": [
  {
    "name": "ML-KEM-768",
    "type": "cryptographic-asset",
    "cryptoProperties": {
      "assetType": "algorithm",
      "algorithmProperties": {
        "algorithmFamily": "ML-DSA",
        "primitive": "kem",
        "executionEnvironment": "software-plain-ram",
        "cryptoFunctions": ["keygen", "encapsulate", "decapsulate"],
        "nistQuantumSecurityLevel": 3
      }
    }
  },
  {
    "name": "x25519",
    "type": "cryptographic-asset",
    "cryptoProperties": {
      "assetType": "algorithm",
      "algorithmProperties": {
        "algorithmFamily": "ECDH",
        "primitive": "key-agree",
        "executionEnvironment": "software-plain-ram",
        "cryptoFunctions": ["keygen", "keyderive"],
        "nistQuantumSecurityLevel": 0
      }
    }
  },
  {
    "name": "ECDH-P-256",
    "type": "cryptographic-asset",
    "cryptoProperties": {
      "assetType": "algorithm",
      "algorithmProperties": {
        "algorithmFamily": "ECDH",
        "primitive": "key-agree",
        "executionEnvironment": "software-plain-ram",
        "cryptoFunctions": ["keygen", "keyderive"],
        "nistQuantumSecurityLevel": 0
      }
    }
  },
  {
    "name": "openssl-config",
    "type": "data",
    "data": {
      "bom-ref": "config-001",
      "type": "configuration",
      "url": "/etc/openssl/ssl/openssl.cnf",
      "contents": {
        "attachment": {
          "contentType": "text/plain",
          "encoding": "utf8",
          "content": "DEFAULT_GROUPS    = x25519:SecP256r1MLKEM768"
        }
      }
    }
  },
  "dependencies": [
    {
      "ref": "ML-KEM-768",
      "dependsOn": ["openssl-config"]
    },
    {
      "ref": "x25519",
      "dependsOn": ["openssl-config"]
    },
    {
      "ref": "ECDH-P-256",
      "dependsOn": ["openssl-config"]
    }
  ]
]
```

Hardware BOM

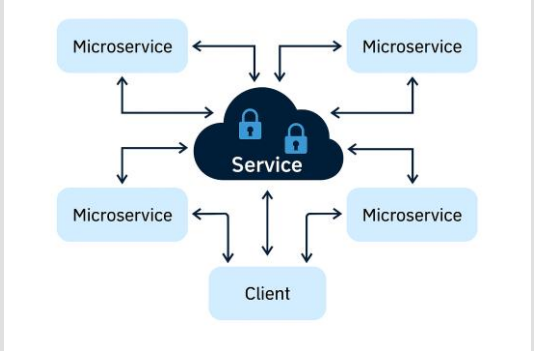


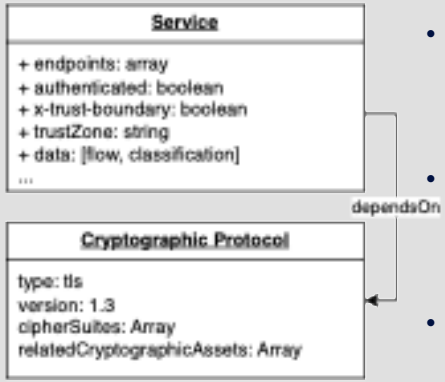
- HBOM models physical components linked to CBOM
- Hardware devices provide cryptography, algorithms, keys

Hardware BOM and **SaaSBOM**



- HBOM models physical components linked to CBOM
- Hardware devices provide cryptography, algorithms, keys





- Model endpoints, data flows and classifications
- Associate protocols, certificates and key material to services
- Helps with compliance and threat modeling

CycloneDX

Manufacturing BOM

Formulation

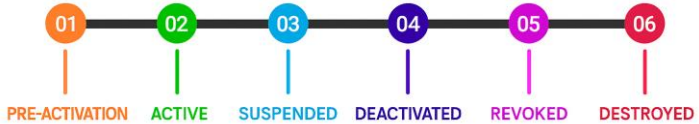
- Capture how components were formed: tasks, inputs, outputs, triggers, steps, runtime
- References components (e.g., cryptographic assets), services, workflows
- Examples for cryptography:
  - Use MBOM to document test procedure
  - Certification workflows

CycloneDX 1.7 new features

improved key and **certificate management**

- Support for key management states following guidelines from NIST SP 800-57

- Intersecting with SDLC life cycle states supported by CycloneDX.
  - Design, Pre-build, Build, Post-Build, Operations, Discovery, Decommission
- Support for certificate lifecycle stages introduced in 1.7, and certificate extensions




```
"components": [
    {
      "name": "revoked-internal-ca.example.com",
      "type": "cryptographic-asset",
      "bom-ref": "840ADC47-55CD-44C6-A306-B37A9149B066",
      "cryptoProperties": {
        "assetType": "certificate",
        "certificateProperties": {
          "serialNumber": "ABCDEF1234567890FEDCBA",
          "subjectName": "CN = internal-ca.example.com, OU = IT Security, O = Example
          "issuerName": "CN = Example Root CA, O = Example Corp, C = US",
          "notValidBefore": "2023-01-01T00:00:00Z",
          "notValidAfter": "2025-12-31T23:59:59Z",
          "certificateFormat": "X.509",
          "certificateFileExtension": "pem",
          "fingerprint": {
            "alg": "SHA-256",
            "content": "9f86d081884c7d659a2feaa0c55ad015a3bf4f1b2b0b822cd15d6c15b0f00a
          },
          "certificateState": [
            {
              "state": "revoked",
              "reason": "Certificate was compromised due to private key exposure in se
            }
          ],
          "creationDate": "2022-12-15T10:00:00Z",
          "activationDate": "2023-01-01T00:00:00Z",
          "revocationDate": "2024-01-10T15:45:30Z",
          "certificateExtensions": [
            {
              "commonExtensionName": "basicConstraints",
              "commonExtensionValue": "CA:TRUE, pathlen:2"
            },
            {
              "commonExtensionName": "keyUsage",
              "commonExtensionValue": "Certificate Sign, CRL Sign, Digital Signature"
            },
            {
              "commonExtensionName": "extendedKeyUsage",
              "commonExtensionValue": "TLS Web Server Authentication, TLS Web Client A
            },
            {
              "commonExtensionName": "subjectAlternativeName",
              "commonExtensionValue": "DNS:internal-ca.example.com, DNS:ca.internal.ex
            },
            {
              "commonExtensionName": "authorityKeyIdentifier",
              "commonExtensionValue": "keyid:01:02:03:04:05:06:07:08:09:0A:0B:0C:0D:0E
            },
            {
              "commonExtensionName": "subjectKeyIdentifier",
              "commonExtensionValue": "A1:B2:C3:D4:E5:F6:07:08:09:0A:0B:0C:0D:0E:0F:10
            },
            {
              "commonExtensionName": "crlDistributionPoints",
              "commonExtensionValue": "URI:http://crl.example.com/root-ca.crl"
            },
            {
              "commonExtensionName": "authorityInformationAccess",
```

Resolving Naming ambiguities

With CycloneDX 1.7

Challenges:

Found	Caveats
Triple-DES	Also known as: DESde, 3DES
Diffie-Hellman	FFDH or ECDH, which elliptic curve
RSA	Signature, PKE or KEM? RSAES OAEP or PKCS#1.5 Or RSASSA-PSS, but which digest, salt and key length
ML-DSA	Pure or HashML-DSA, which parameter set?

- Multiple names for the same algorithm
- Details needed

CycloneDX 1.7 introduces algorithm definitions with:

- Algorithm families (e.g., RSASSA-PKCS1)
- **Naming patterns** to unify synonyms
- Coverage driven by real-world use cases:
  - TLS, IPSEC, PKCS#11, Telco/5G profiles, and further algorithms
- Elliptic curve definitions, with synonyms *

```
"algorithms": [
    {
      "family": "RSASSA-PKCS1",
      "standard": [
        {"name": "RFC8017", "url": "https://doi.org/10.17487/RFC8017"},
        {"name": "IEEE1363", "url": "https://doi.org/10.1109/IEEESTD.2000.92290"}
      ],
      "variant": [
        {
          "pattern": "RSA-PKCS1-1.5[-{digestAlgorithm}][-{keyLength}]",
          "primitive": "signature"
        }
      ]
    }
]
```

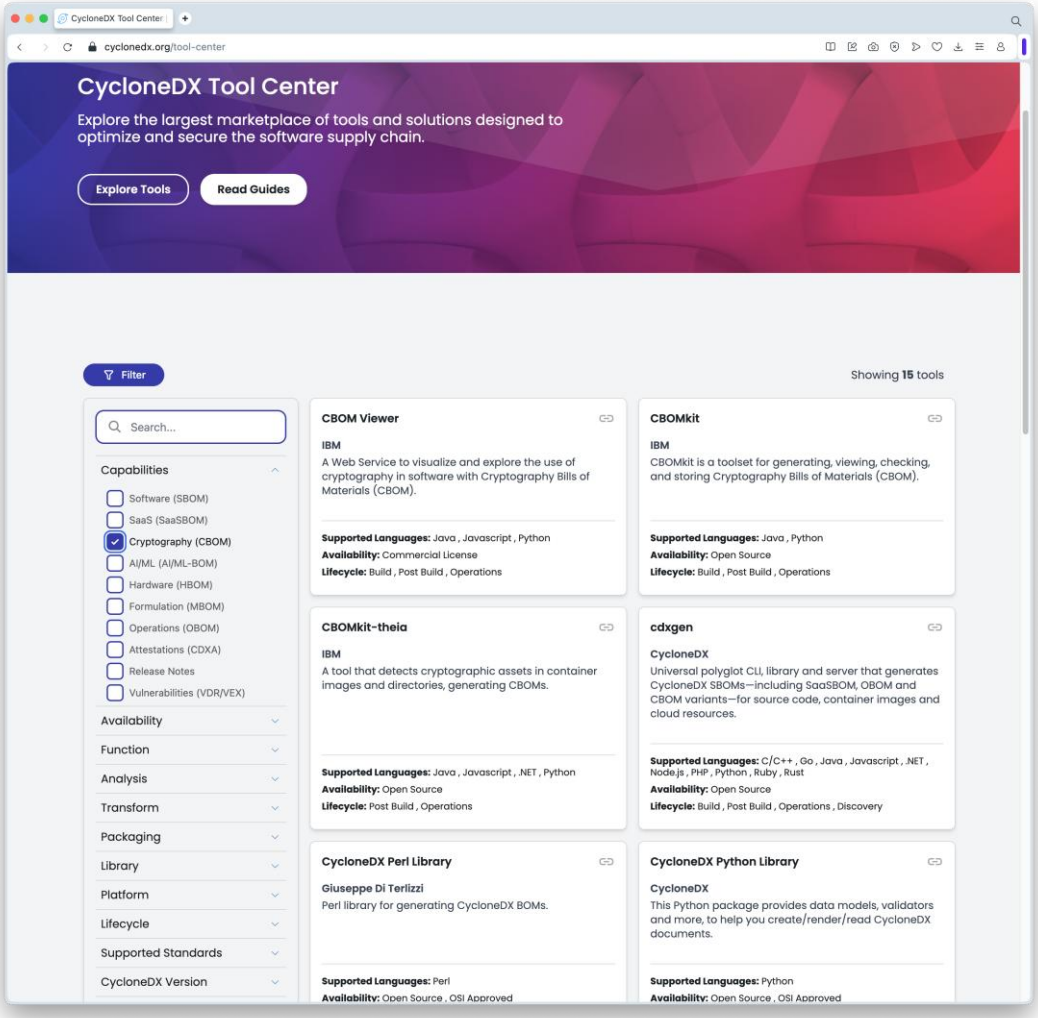
* derived from std-curves (CRoCS) (https://github.com/J08nY/std-curves)

CBOM Tooling

CycloneDX Tool Center

Open Source and Commercial tooling related to (C)BOM

https://cyclonedx.org/tool-center

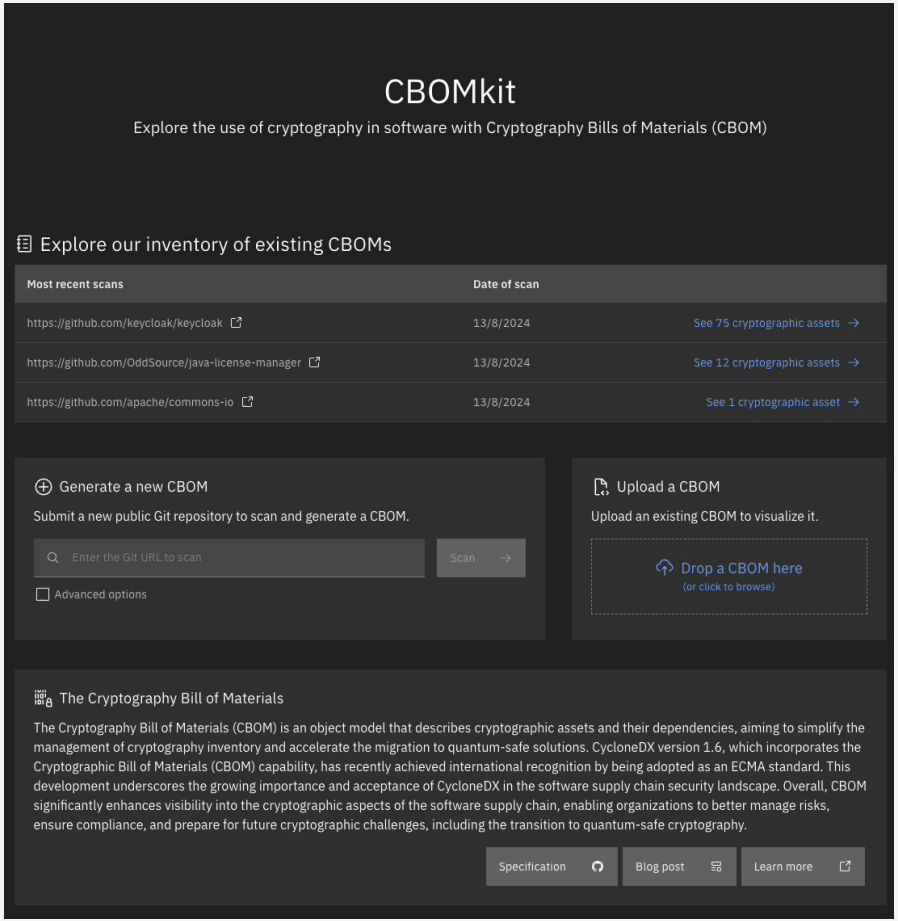


CBOM Tooling: CBOMkit

CBOMkit is an open-source toolset for dealing with Cryptography Bill of Materials (CBOM).

- **CBOM Generation (CBOMkit-hyperion, CBOMkit-theia):** Generate CBOMs from source code by scanning git repositories to find the used cryptography.
- **CBOM Viewer (aka CBOMkit-coeus):** Visualize a generated or uploaded CBOM and access comprehensive statistics.
- **CBOM Compliance Check:** Evaluate CBOMs created or uploaded against specified compliance policies and receive detailed compliance status reports.
- **CBOM Database:** Collect and store CBOMs into the database and expose this data through a RESTful API

https://github.com/PQCA/cbomkit



CBOM Compliance Checks

Common Expression Language (CEL)


```
"components": [
  {
    "name": "desede-168-cbc-pkcs5",
    "type": "cryptographic-asset",
    "bom-ref": "55de8502-da48-4e77-b130-b852b54940b7",
    "cryptoProperties": {
      "assetType": "algorithm",
      "algorithmProperties": {
        "padding": "pkcs5",
        "primitive": "block-cipher",
        "cryptoFunctions": [
          "decrypt"
        ],
        "parameterSetIdentifier": "168",
        "nistQuantumSecurityLevel": 1
      }
    },
    "evidence": {
      "occurrences": [
        {
          "line": 332,
          "offset": 36,
          "location": "java/org/apache/tomcat/util/net/jsse/PEMFile.java",
          "additionalContext": "javax.crypto.Cipher#getInstance(Ljava/lang/String;)Ljavax/crypto/Cipher;"
        }
      ]
    }
  },
```

CBOM Compliance Checks

Common Expression Language (CEL)

"No md5 usage"

```
components.all(c,
        c.assetType == "algorithm" ?
        c.name != "md5" :
        true)
```


```
"components": [
  {
    "name": "desede-168-cbc-pkcs5",
    "type": "cryptographic-asset",
    "bom-ref": "55de8502-da48-4e77-b130-b852b54940b7",
    "cryptoProperties": {
      "assetType": "algorithm",
      "algorithmProperties": {
        "padding": "pkcs5",
        "primitive": "block-cipher",
        "cryptoFunctions": [
          "decrypt"
        ],
        "parameterSetIdentifier": "168",
        "nistQuantumSecurityLevel": 1
      }
    },
    "evidence": {
      "occurrences": [
        {
          "line": 332,
          "offset": 36,
          "location": "java/org/apache/tomcat/util/net/jsse/PEMFile.java",
          "additionalContext": "javax.crypto.Cipher#getInstance(Ljava/lang/String;)Ljavax/crypto/Cipher;"
        }
      ]
    }
  },
```

CBOM Compliance Checks

Common Expression Language (CEL)

```
"components": [
  {
    "name": "desede-168-cbc-pkcs5",
    "type": "cryptographic-asset",
    "bom-ref": "55de8502-da48-4e77-b130-b852b54940b7",
    "cryptoProperties": {
      "assetType": "algorithm",
      "algorithmProperties": {
        "padding": "pkcs5",
        "primitive": "block-cipher",
        "cryptoFunctions": [
          "decrypt"
        ],
        "parameterSetIdentifier": "168",
        "nistQuantumSecurityLevel": 1
      }
    },
    "evidence": {
      "occurrences": [
        {
          "line": 332,
          "offset": 36,
          "location": "java/org/apache/tomcat/util/net/jsse/PEMFile.java",
          "additionalContext": "javax.crypto.Cipher#getInstance(Ljava/lang/String;)Ljavax/crypto/Cipher;"
        }
      ]
    }
  },
```

"No md5 usage"

```
components.all(c,
        c.assetType == "algorithm" ?
        c.name != "md5" :
        true)
```

"The key size for RSA should be greater or equal to 2048"

```
components.all(c,
        c.assetType == "algorithm" ?
        (c.name.contains("rsa") && c.parameterSetIdentifier >= 2048)) :
        true)
```

Conclusion

- **Inventories and CBOM become a must** for transparency and compliance
- **CBOM goes beyond software:** covers services, configuration and workflows
- **CycloneDX** provides an interchangeable standard for supply chains
- **Tooling is maturing**: CBOMkit and CycloneDX ecosystem make adoption practical.

Future outlook

CycloneDX 2.0 plans

- Blueprints and Bill of Behaviors
  - CBOM use case: SHA-3 in a CBOM appears secure. But what if the purpose of the software is password hashing
- Risk modeling
  - Capture threats, modelling the quantum threat, including mitigations and mitigation roadmaps

THANK YOU