Post-Quantum

Cryptography Conference

Is CBOM Enough?

A number of organizations are framing the migration to post-quantum cryptography (PQC) as an opportunity to affect broader IT security modernization across their organizations, especially as it relates to managing the full life cycle of cryptographic algorithms, libraries, and protocols. This presentation introduces the idea of a Cryptographic Bill of Materials (CBOM), often considered a key element of this IT modernization effort. We compare and contrast CBOM with the more familiar Software Bill of Materials (SBOM), paying particular attention to how the cryptographic ecosystem poses unique challenges compared to software-at-large. For example, many cryptographic protocols include a negotiation phase over the wire, complicating the effort to know exactly which algorithm was used in any given protocol handshake. We explore the types of cryptographic assurance that a CBOM can and cannot provide for an organization. Finally, we describe how supplementary efforts such as key management, real-time cryptographic monitoring, and the ability to execute historical queries are needed to fill in the operational gaps of a CBOM.

Roman Cinkais
Co-founder at 3Key Company

SSL.com PQ SHIELD HID KEYFACTOR ENTRUST

January 15 and 16, 2025 - Austin, TX (US) | Online

PKI Consortium

Is CBOM enough?

16.1.2025

PKI Consortium

Post-Quantum Cryptography Conference

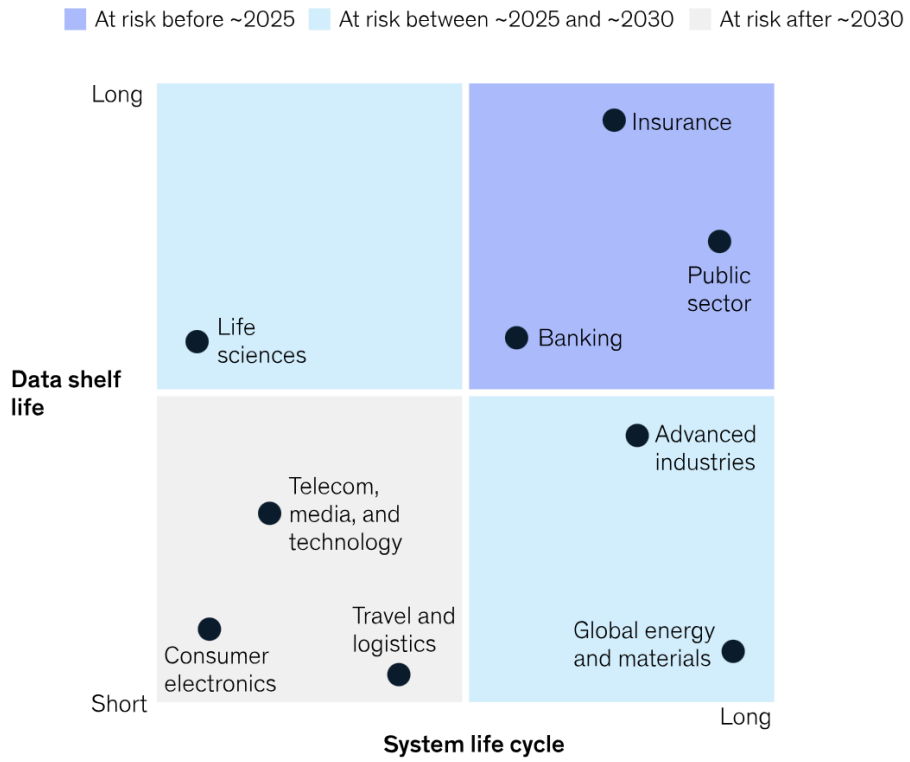
January 15 and 16, 2025 - Austin, Texas, US

Roman Cinkais

3KEYCOMPANY

URGENCY

Risk of quantum-powered attack by industry



McKinsey & Company

- **Data Shelf Life**: How long sensitive data must remain secure.
- **System Life and Development Cycles**: The timeline of system updates and improvements.

Industries such as finance, insurance, and government face particularly urgent challenges. These sectors manage cryptographic assets with long-term lifecycles, and any delay in implementing PQC could result in the compromise of highly sensitive data once quantum computing becomes mainstream.

- NIST - **Transition to Post-Quantum Cryptography Standards**
- A joint statement from partners from 18 EU member states - **Securing Tomorrow, Today: Transitioning to PostQuantum Cryptography**
- Deprecation of RSA and ECDSA after 2030

PREPARE FOR TRANSITION

Building PQC Awareness

Discovery and Testing of Implementations

Inventory of Cryptographic Assets

Adopting Cryptographic Agility

Risk Assessment and Migration Decision

Managing the Migration

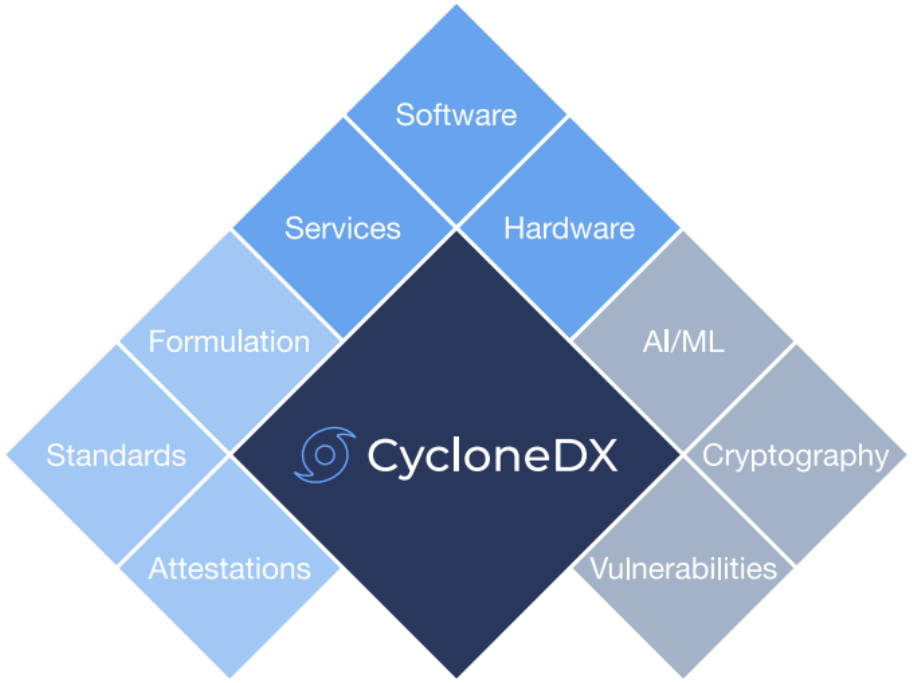
- **Legacy systems will need to be replaced or put into quantum-safe wrapper**
- **It will be hard without having interoperability and standards**
- **How long it will take? Are we able to make it by 2035?**
- **What next?**

xBOM (Bill of Materials)

- BOM is a machine-readable document that provides a comprehensive inventory of components, dependencies, and their associated metadata for a software application or system.
- Focused on improving software transparency, mitigating supply chain risks, and ensuring security including regulatory compliance.
- Primary machine-readable formats are SPDX and CycloneDX, however can be expresses in any format

CYCLONEDX

- OWASP CycloneDX is a full-stack Bill of Materials (BOM) standard that provides advanced supply chain capabilities for cyber risk reduction. CycloneDX is an Ecma International standard published as ECMA-424. The OWASP Foundation and Ecma International Technical Committee for Software & System Transparency (TC54) drive the continued advancement of the specification.



- Software Bill of Materials (SBOM)
- Software-as-a-Service Bill of Materials (SaaSBOM)
- Hardware Bill of Materials (HBOM)
- Machine Learning Bill of Materials (ML-BOM)
- Cryptography Bill of Materials (CBOM)
- Operations Bill of Materials (OBOM)
- Manufacturing Bill of Materials (MBOM)
- Bill of Vulnerabilities (BOV)
- Vulnerability Disclosure Report (VDR)
- Vulnerability Exploitability eXchange (VEX)
- CycloneDX Attestations (CDXA)

SBOM

- *SBOMs describe the inventory of software components and services and the dependency relationships between them. A complete and accurate inventory of all first-party and third-party components is essential for risk identification. SBOMs should ideally contain all direct and transitive components and the dependency relationships between them.*

- NTIA - The Minimum Elements For a Software Bill of Materials (SBOM) Pursuant to Executive Order 14028 on Improving the Nation's Cybersecurity (2021)

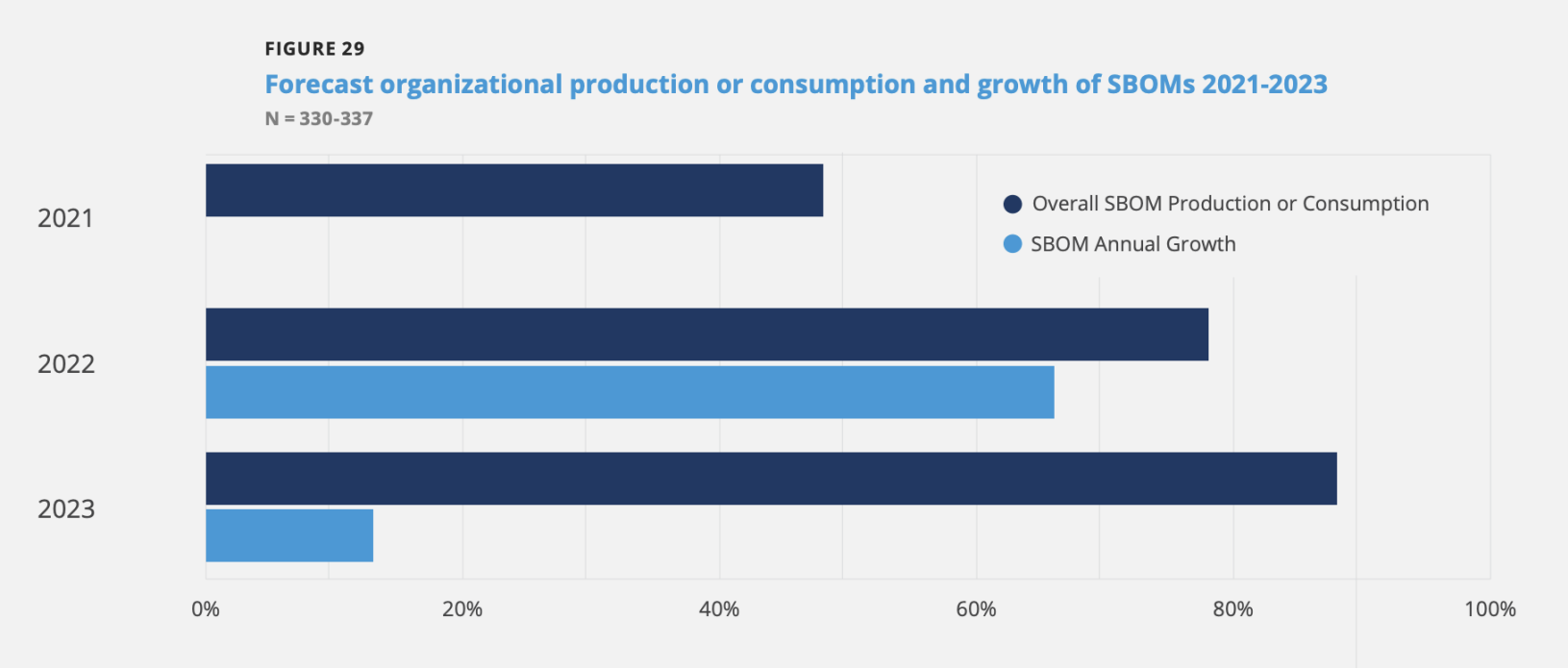
```
{
 "bomFormat": "CycloneDX",
 "specVersion": "1.2",
 "serialNumber": "urn:uuid:411dafd2-c29f-491a-97d7-e97de5bc2289",
 "version": 1,
 "metadata": {
  ...
 },
 "components": [
  ...
 ],
 "dependencies": [
  {"ref": "pkg:maven/org.bouncycastle/bcprov-jdk15on@1.62?type=jar"},
  ...
 ]
}
```


SBOM

- *The Linux Foundation - The State of Software Bill of Materials (SBOM) and Cybersecurity Readiness from 2022*
- **“SBOMs are no longer optional”**

- 76% of organizations have a degree of SBOM “readiness”
- 47% were actively using SBOMs in 2021

- This figure is predicted to rise to over three-quarters of organizations in 2022 (78%) and almost 90% the following year.

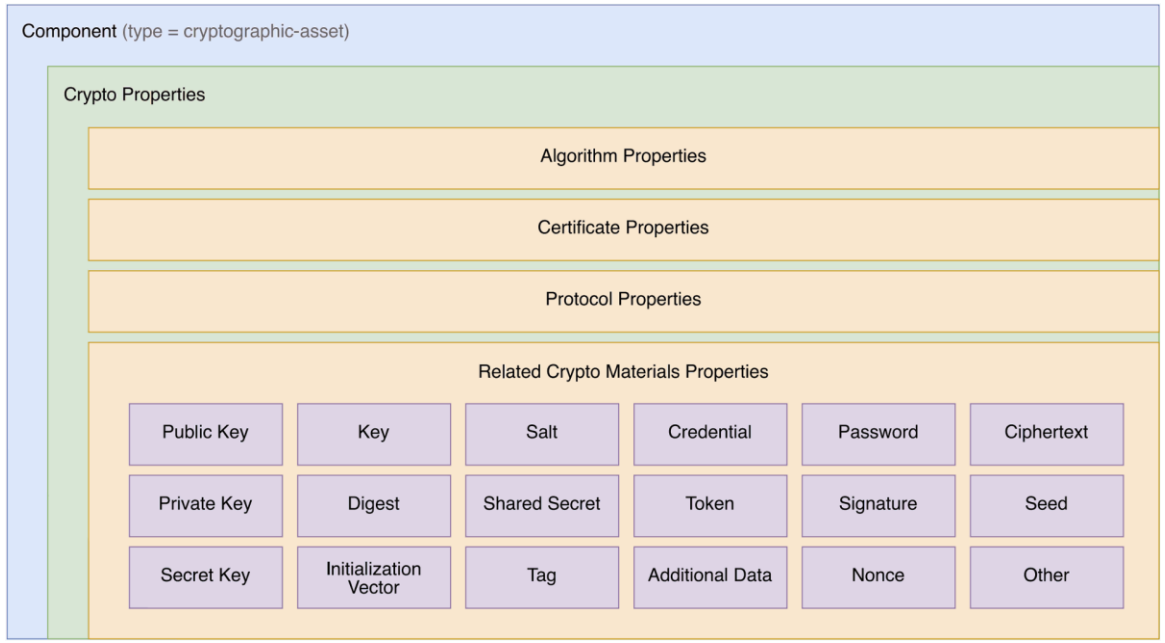


CBOM

To streamline and standardize the inventory process, many organizations are turning to the Cryptography Bill of Materials (CBOM). The CBOM is an extensible model that allows organizations to describe cryptographic assets and their dependencies in a structured format. It is part of the OWASP CycloneDX framework, which is designed to reduce cyber risks by providing a detailed picture of cryptographic infrastructure.

- **Modeling Cryptographic Assets**: CBOM provides a framework to model cryptographic assets, including keys, certificates, algorithms, and protocols.
- **Capturing Asset Properties**: It captures key attributes such as algorithm type, key length, usage details, and more.
- **Tracking Dependencies**: CBOM maps out dependencies between different cryptographic assets and the systems that use them.
- **Compatibility with CycloneDX**: The CBOM is fully compatible with the CycloneDX Software Bill of Materials (SBOM) standard, making it easier to integrate into existing supply chain management tools.

CBOM (+SBOM)



CBOM KEY


```
"components": [
 {
  "name": "RSA-2048",
  "type": "cryptographic-asset",
  "bom-ref": "crypto/key/rsa-2048@1.2.840.113549.1.1.1",
  "cryptoProperties": {
   "assetType": "related-crypto-material",
   "relatedCryptoMaterialProperties": {
    "type": "public-key",
    "id": "2e9ef09e-dfac-4526-96b4-d02f31af1b22",
    "state": "active",
    "size": 2048,
    "algorithmRef": "crypto/algorithm/rsa-2048@1.2.840.113549.1.1.1",
    "securedBy": {
     "mechanism": "Software",
     "algorithmRef": "crypto/algorithm/aes-128-gcm@2.16.840.1.101.3.4.1.6"
    },
    "creationDate": "2016-11-21T08:00:00Z",
    "activationDate": "2016-11-21T08:20:00Z"
   },
   "oid": "1.2.840.113549.1.1.1"
  }
 },...
```


CBOM ALGORITHM


```
"components": [
 {
  "name": "Dilithium5",
  "type": "cryptographic-asset",
  "cryptoProperties": {
   "assetType": "algorithm",
   "algorithmProperties": {
    "primitive": "signature",
    "executionEnvironment": "software-plain-ram",
    "implementationPlatform": "x86_64",
    "certificationLevel": [
     "none”
    ],
    "cryptoFunctions": [
     "keygen",
     "sign",
     "verify”
    ],
    "nistQuantumSecurityLevel": 5
   },
   "oid": "1.3.6.1.4.1.2.267.7.8.7"
  }
 }
]
```


CBOM CERTIFICATE


```
"components": [
 {
  "name": "google.com",
  "type": "cryptographic-asset",
  "bom-ref": "crypto/certificate/google.com@sha256:1e15...beb4",
  "cryptoProperties": {
   "assetType": "certificate",
   "certificateProperties": {
    "subjectName": "CN = www.google.com",
    "issuerName": "C = US, O = Google Trust Services LLC, CN = GTS CA
1C3",
    "notValidBefore": "2016-11-21T08:00:00Z",
    "notValidAfter": "2017-11-22T07:59:59Z",
    "signatureAlgorithmRef": "crypto/algorithm/sha-512-
rsa@1.2.840.113549.1.1.13",
    "subjectPublicKeyRef": "crypto/key/rsa-2048@1.2.840.113549.1.1.1",
    "certificateFormat": "X.509",
    "certificateExtension": "crt"
   }
  }
 },...
```

```
...
{
 "name": "SHA512withRSA",
 "type": "cryptographic-asset",
 "bom-ref": "crypto/algorithm/sha-512-rsa@1.2.840.113549.1.1.13",
 "cryptoProperties": { ... }
},...
```

```
...
{
 "name": "RSA-2048",
 "type": "cryptographic-asset",
 "bom-ref": "crypto/key/rsa-2048@1.2.840.113549.1.1.1",
 "cryptoProperties": { ... }
},...
```


SBOM + CBOM

- BOM-Link in format `urn:cdx:serialNumber/version#bom-ref`

```
urn:cdx:f08a6ccd-4dce-4759-bd84-c626675d60a7/1
urn:cdx:f08a6ccd-4dce-4759-bd84-c626675d60a7/1#componentA
```

- External references point to resources outside the object they're associated with and may be external to the BOM, or may refer to resources within the BOM

```
"components": [
 {
  "type": "application",
  "name": "Acme Application",
  "version": "1.0.0",
  "externalReferences": [
   {
    "type": "bom",
    "url": "https://example.com/bom/acme-application-1.0.0-cbom.cdx.json",
    "hashes": [
     {
      "alg": "SHA-256",
      "content":
"708f1f53b41f11f02d12a11b1a38d2905d47b099afc71a0f1124ef8582ec7313"
     }
    ]
   }
  ]
 }
]
```


EXTENSIBILITY

- There are three primary means of extending BOM
  - **BOM properties** - name-value store that can be used to describe additional data about the components, services, or the BOM that isn't native to the core specification

```
"properties": [
 {
  "name": "Foo",
  "value": "Bar"
 }
]
```

  - **BOM properties using registered namespace** - hierarchical and delimited with a `:` and may optionally start with `urn:`. Should be registered using taxonomy repository

```
"properties": [
 {
  "name": "snyk:org_id",
  "value": "2473c3d3-d161-4c4e-acc6-8ec476419645"
 }
]
```

  - **XML extensions** - additional XML elements so long as they reside in a different namespace, allows for representing more complex data structures, for example enveloped signature

WHAT IS INVENTORY

Cryptographic Keys

These are the fundamental building blocks used for encryption and decryption. For each key, details such as storage type, ownership, algorithm identifier, format, and status (active or deprecated) should be documented.

Certificates

Certificates serve as electronic identities, often tied to public keys used in encryption. Information such as validity period, ownership, and algorithm used should be captured.

Algorithms

Different cryptographic algorithms serve various functions, such as data encryption, authentication, and digital signatures. It is important to track which algorithms are in use, especially to identify algorithms that are vulnerable to quantum attacks.

Protocols

Protocols such as TLS or IPSec specify how cryptographic operations should be carried out between two or more entities. Protocol version and implementation details should be part of the inventory to track any dependencies.

BUILDING INVENTORY

1

The first step in building an inventory involves identifying where cryptographic assets are used across your infrastructure. These assets could be stored in:

- Certification Authorities (CAs)
- Servers and Databases
- Hardware Security Modules (HSMs)
- Network traffic, including encrypted traffic
- Cloud providers
- Containers and microservices

Given the complex and hybrid nature of modern IT environments, relying on manual methods like spreadsheets is impractical. Instead, automated tools and software solutions that support discovery across diverse platforms are essential.

2

Once discovered, you need to gather metadata about these cryptographic assets. This includes:

- Ownership: Who is responsible for managing the cryptographic asset?
- Application: What applications or systems are using the asset?
- Dependencies: What other systems, libraries, or services depend on the asset?

Understanding these details ensures that organizations have the necessary context to manage risks, particularly as they transition to post-quantum cryptography.

3

A well-structured inventory provides several benefits:

- Planning: It helps identify outdated algorithms and cryptographic assets that need to be replaced or upgraded.
- Risk Assessment: Regular monitoring of assets over time allows for proactive risk management, ensuring compliance and tracking progress toward quantum readiness.
- Identifying Dependencies: Knowing how cryptographic assets are interconnected can prevent system outages during the migration to PQC.

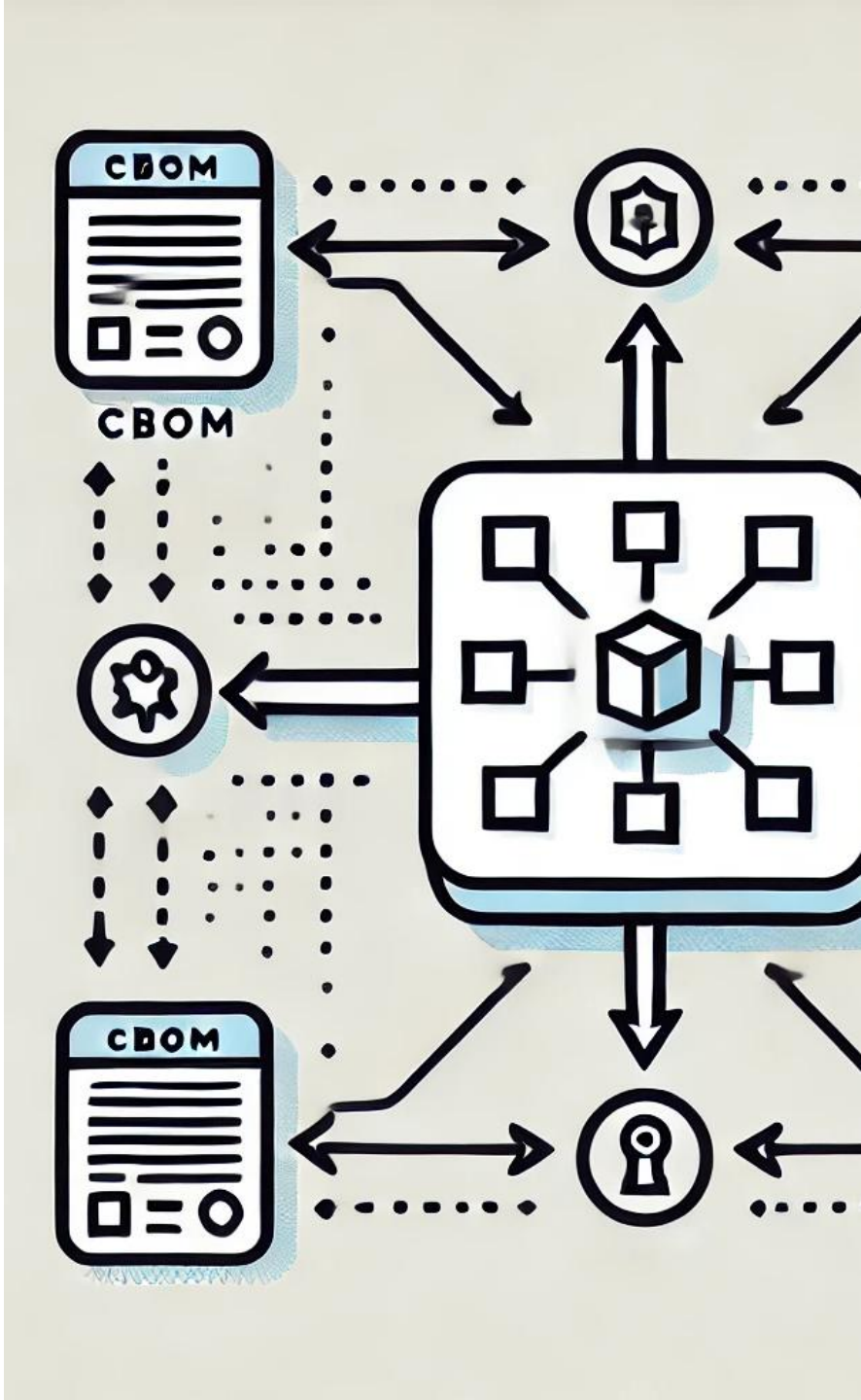
VISIBILITY

- Identification of cryptography assets through:
  - **Manual** interpretation
  - **Static** analysis and scans
  - **Dynamic** analysis of operations

Consolidation of Inventory:
  - Different CBOMs that need to be consolidated to provide one consistent overview
  - Keeping the relationships between cryptography assets for analysis

Automation of Discovery Process:
  - Frequent or real-time collection of cryptography usage?
  - Management of changes

Adoption of Telemetry:
  - Unavailability of information from legacy systems
  - Adoption of standards and open-source tools like OpenTelemetry for better visibility



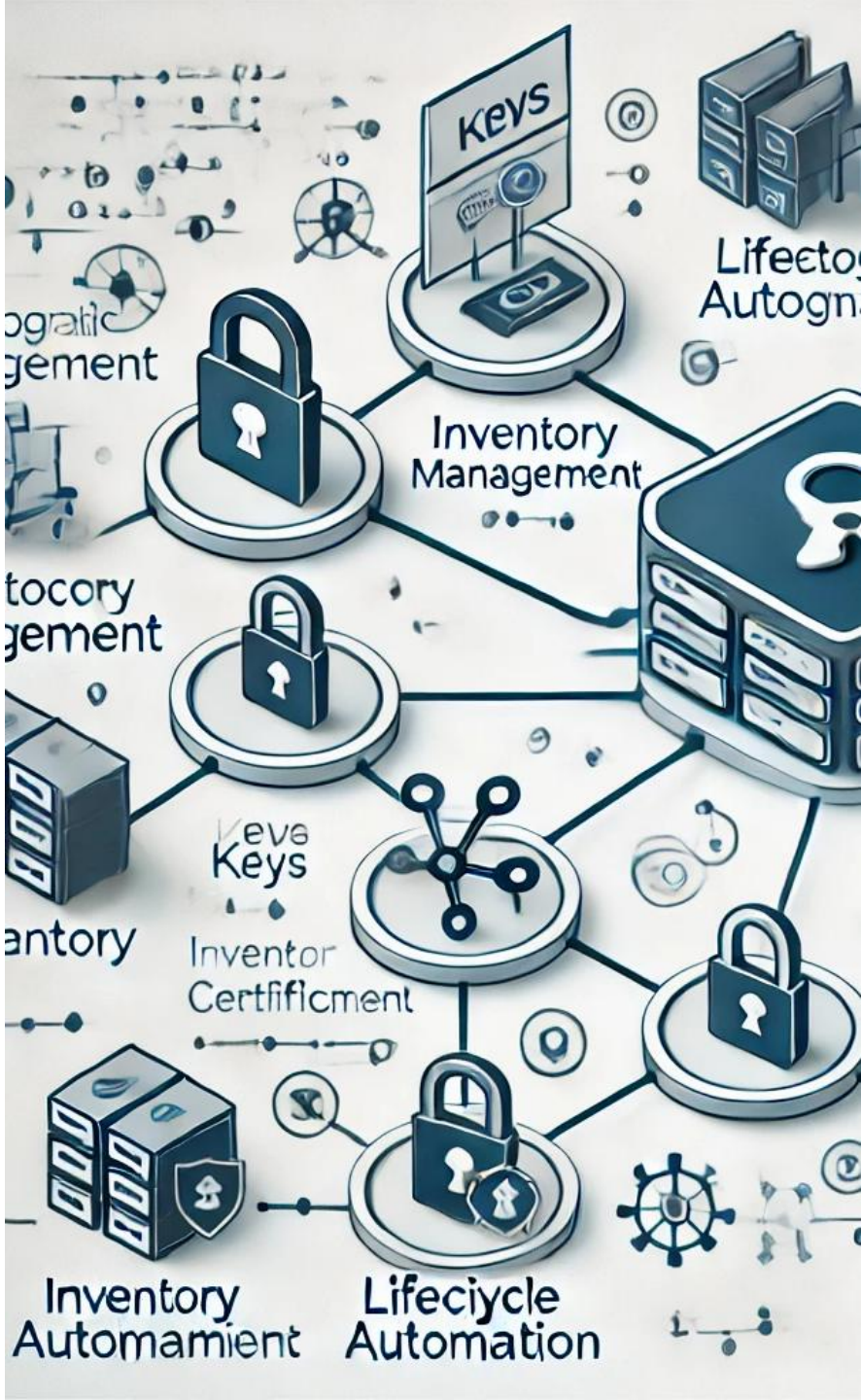
MANAGEMENT

- Proper management ensures:
  - **Agility**: Seamless adaptation to new cryptographic standards like PQC
  - **Compliance**: Meeting regulatory and industry requirements
  - **Risk Mitigation**: Preventing vulnerabilities from legacy cryptography

Lifecycle Management:
  - Automate key and certificate renewal processes
  - Ensure timely deprecation of outdated algorithms and keys

Policy Enforcement:
  - Enforce organization-wide cryptographic policies (e.g., minimum key lengths)
  - Regular audits for compliance and standards adherence

Interoperability Across Systems:
  - Integrate cryptographic management tools with hybrid IT environments
  - Ensure compatibility with legacy and modern infrastructures



MONITORING

- Provides real-time visibility into cryptographic usage
- Detects vulnerabilities in legacy systems and insecure cryptography
- Ensures compliance with policies and standards dynamically

Network Monitoring:
  - Analyze over-the-wire cryptographic activity
  - Example: CryptoMon to inspect TLS protocols and generate CBOMs

Real-Time Monitoring:
  - Continuous tracking of cryptographic asset usage
  - Identifies deviations from approved algorithms or protocols

Historical Analysis:
  - Enables post-event analysis and forensic investigations
  - Complements real-time tracking for a comprehensive view.

- Difficulty identifying cryptographic assets in hybrid systems
- Lack of standardized tools for historical and real-time queries
- False positives due to outdated libraries or misconfigurations

LIMITATIONS

Lack of Real-Time Monitoring

CBOM provides a static inventory and cannot reflect dynamic changes in cryptographic usage. Real-time cryptographic activity monitoring tools are required for operational visibility.

Limited Adoption and Coverage

CBOM is still in its early stages compared to SBOM, with low industry adoption and standardization. Organizations face challenges in achieving full inventory coverage.

No Automated Key Management

CBOM lacks built-in mechanisms to manage cryptographic keys across their lifecycle. Dedicated key management solutions are necessary for secure handling.

Legacy Cryptography

Identifying and replacing legacy or insecure cryptographic implementations can be difficult. False positives in asset discovery can hinder effective migration to post-quantum cryptography.

Static Nature

CBOM focuses on documentation but lacks tools for proactive enforcement of cryptographic policies. Historical and forensic analysis capabilities are absent.

Interoperability Challenges

Integration with hybrid environments, cloud services, and diverse infrastructure requires significant effort. Open-source tools and frameworks like CycloneDX can help but are not fully adopted.

CONSIDERATIONS

Challenges

False positives in cryptographic discovery

Legacy libraries with insecure cryptography

Lack of real-time monitoring

Poor adoption and coverage

Static nature of CBOM

Historical queries and post-event analysis

Interoperability with hybrid systems

Solutions

Use advanced inventory tools with validation

Implement strict policies and deprecation

Introduce tools like CryptoMon or similar

Standardize CBOM practices across the org

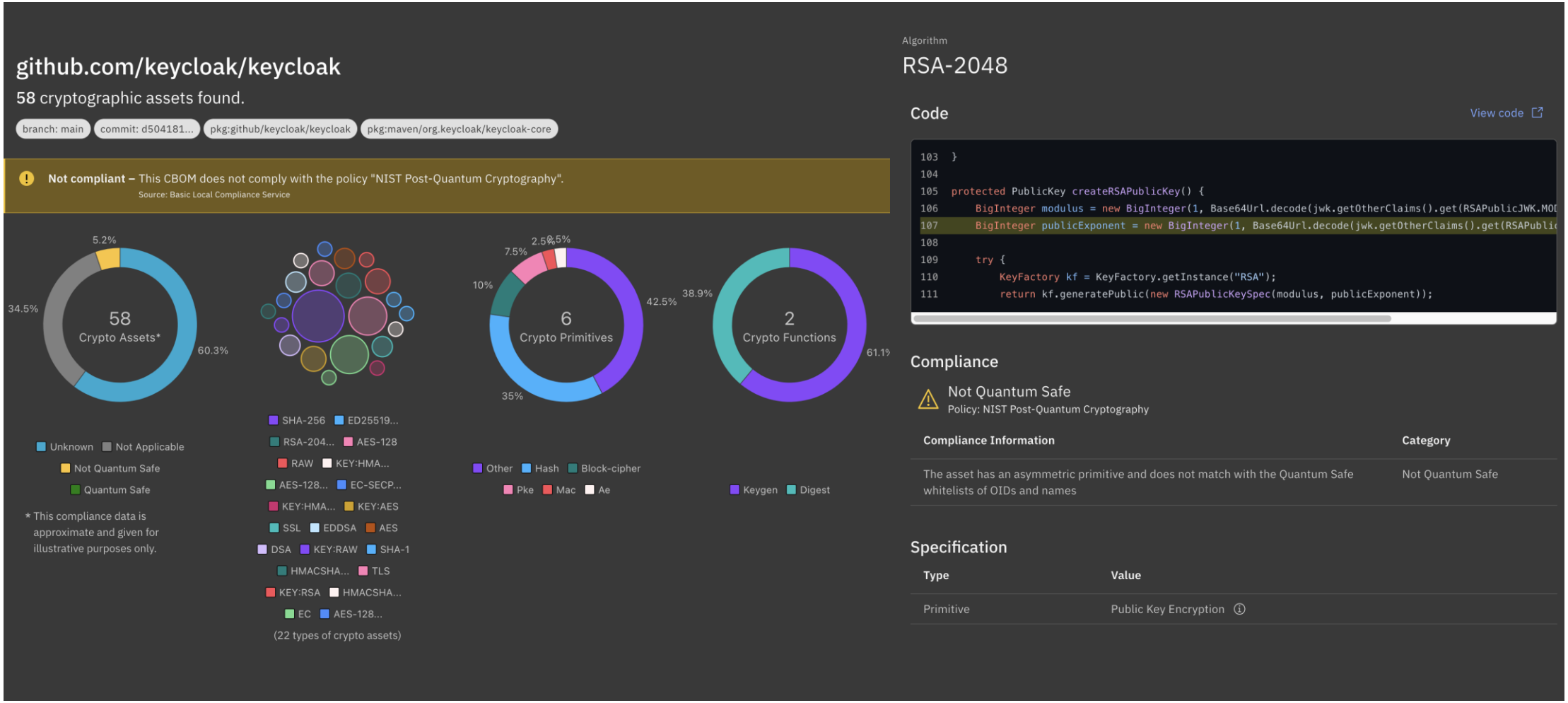
Combine CBOM with dynamic monitoring tools

Utilize supplementary query and analytics

Leverage frameworks like CycloneDX

“Achieving cryptographic security isn’t a sprint—it’s a marathon. CBOM provides the map; it’s up to us to ensure we bring the right gear.”

CBOMKIT



CBOM – SUFFICIENT OR NOT?

- CBOM is a critical first step - but only a step. Let’s complete the journey.
- CBOM is an essential tool for managing cryptographic assets, but it cannot address all operational and security needs alone.

Feature	CBOM	Needs Additional Tools
Comprehensive cryptographic inventory	✅	
Key management and lifecycle tracking		✅
Real-time cryptographic monitoring		✅
Historical queries and post-event analysis		✅
Risk assessments and proactive alerts	✅	
Algorithm and protocol compliance checks	✅	
Handling legacy cryptographic assets		✅
Interoperability across hybrid systems	✅	
Vulnerability reporting and remediation	✅	
Automation for cryptographic discovery	✅	
Reduction of false positives in inventory		✅

3Key Company s.r.o.
Roman Cinkais
roman.cinkais@3key.company