

## **Comment to NIST.CSWP.39.ipd**

---

|                                                    |   |
|----------------------------------------------------|---|
| 1. Introduction.....                               | 1 |
| 2. Summary.....                                    | 2 |
| 3. Comment/Suggestion on Abstract.....             | 3 |
| 4. Comments/Suggestions on Introduction.....       | 3 |
| 5. Comments/Suggestions on Section 3.3.....        | 5 |
| 6. Comments/Suggestions on Section 4.2.....        | 7 |
| 7. Public Comment about System-Level Security..... | 8 |

## **Comment to NIST.CSWP.39.ipd**

---

### **1. Introduction**

This document provides comments to the March 5, 2025 Initial Public Draft of NIST CSWP 39: Considerations for Achieving Crypto Agility

Submitted by: Noel Grover, CTO, EnQuanta

Date: April 30, 2025

To: [crypto-agility@nist.gov](mailto:crypto-agility@nist.gov)

Submission Type: Commentary on Initial Public Draft of Cybersecurity White Paper

NIST Technical Series Publication Citation: Barker E, Chen L, Moody D, Regenscheid A, Souppaya M, Newhouse B, Housley R, Turner S (2025) Considerations for Achieving Crypto Agility: Strategies and Practices. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Cybersecurity White Paper (CSWP) NIST CSWP 39 ipd.

<https://doi.org/10.6028/NIST.CSWP.39.ipd>

### **2. Summary**

EnQuanta strongly supports NIST's leadership in recognizing crypto agility as a vital pillar of modern cybersecurity. We commend NIST's ongoing efforts to provide clear guidance on the motivations, methodologies, challenges, and opportunities associated with building agile cryptographic systems. As the cryptographic landscape continues to evolve—fueled by emerging threats, new standards, and the transition to post-quantum cryptography—NIST's role in establishing a shared foundation for crypto-agile strategies is critical to ensuring secure, resilient infrastructure across sectors.

As an emerging technology company specializing in hybrid cryptography, EnQuanta is developing advanced cryptographic frameworks and secure communication systems designed to address the realities of a dynamic threat environment. Our focus is on modular, policy-driven cryptographic engines that support runtime adaptability and composable entropy models. These capabilities enable secure transitions, minimize protocol fragility, and offer strong resistance to both classical and quantum adversaries. Our hybrid cryptographic framework employs a dynamic combination of standard and non-standard ciphers to generate a unique cryptex for each ciphertext, providing resilience and crypto-agility while ensuring compliance through the inclusion of one or more NIST-approved, mandatory-to-implement algorithms.

## Comment to NIST.CSWP.39.ipd

---

The purpose of this submission is to provide constructive feedback on NIST's Draft White Paper and contribute meaningfully to the broader dialogue around crypto agility. In particular, we aim to expand the scope of the survey of current approaches to crypto agility by sharing insights from our work in hybrid cryptography and dynamic cipher sequencing. We hope this perspective will help inform and refine the recommendations and frameworks proposed in the White Paper.

### 3. Comment/Suggestion on Abstract

**Section:** Abstract – Page i, Lines 2-3

**Comment:**

The reference to the “capabilities” of cryptographic agility as what is needed to “replace and adapt cryptographic algorithms” in the Abstract should be expanded to encompass all variations of replacement and/or adaption of cryptographic ciphers to include potential hybrid cryptographic approaches.

It is suggested the first sentence of the Abstract be revised as follows:

*Cryptographic (crypto) agility* refers to the capabilities needed to replace and/or adapt cryptographic algorithms in protocols, applications, software, hardware, and infrastructures without interrupting the flow of a running system in order to achieve resiliency.

### 4. Comments/Suggestions on Introduction

**Section:** 1. Introduction – Page 1, Lines 76-78

**Comment:**

The threats to existing cryptography are not just limited to CRQCs with respect to public-key cryptography. Additional threats such as Quantum (computing and algorithms) enhanced with Artificial Intelligence (QAI) attacks against PQC algorithms may be arising more quickly than previously expected.

It is suggested the sentence at lines 76-78 be revised as follows:

For example, the threats posed by future cryptographically relevant quantum computers (CRQCs) and CRQCs enhanced with Artificial Intelligence (QAI) attacks to public-key cryptography demand an urgent migration to quantum-resistant cryptography.

## Comment to NIST.CSWP.39.ipd

---

**Section:** 1. Introduction – Page 1, Line 93

**Comment:**

The sentence refers to “the proposed definition” but it is unclear which of three different definitions is being referenced: (i) the definition presented in the Abstract at lines 2-4, (ii) the definition presented in the Introduction at lines 80-82, or (iii) the definition from the 2016 NIST presentation [1] set forth in the 3 bullet points at lines 87-92.

It is assumed that the sentence is referring to the definition from the 2016 NIST presentation, but clarification is requested.

**Section:** 1. Introduction – Page 1, Lines 94-96

**Comment:**

The second sentence in this paragraph does a good job of focusing not just on the results of having crypto agility, but also on some of the properties of a cryptographic framework having crypto agility.

It is suggested that these ideal properties could be expanded in a summary fashion by revising the sentence at lines 94-96 as follows:

Crypto agility facilitates migrations between cryptographic algorithms without significant changes to the application that is using the algorithms such that a cryptographic framework ideally supports quantum-resilient algorithms, and is algorithm-adaptable, data-agnostic, backward-compatible and forward-migratable.

**Section:** 1. Introduction – Page 1, Lines 99-101

**Comment:**

The discussion of implementation and replacement in these first two sentences seems unnecessarily limiting. For example, in hybrid cryptographic frameworks utilizing a set of algorithms, the change may only be to limit forward-use of a single weak algorithm in the set of ciphers or algorithms.

It is suggested the sentence at lines 99-101 be revised as follows:

Cryptographic algorithms are implemented in software and/or hardware to facilitate their use in applications. For example, replacing, and/or limiting a cryptographic algorithm in applications ~~will~~may require changes to application programming interfaces (APIs) and or updates to software libraries/modules/interfaces.

---

## Comment to NIST.CSWP.39.ipd

---

**Section:** 1. Introduction – Page 2, Lines 111-117

**Comment:**

The discussion of how to migrate from weak algorithms to new algorithms in this sentence seems unnecessarily limiting and could be interpreted to exclude limiting use of a weak algorithm and using other existing algorithms instead of adopting a new algorithm. These kinds of modification are possible in a hybrid cryptographic framework with a cipher suite that includes a set of two or more ciphers.

It is suggested the sentence at lines 111-117 be revised as follows:

Organizations that practice crypto agility should be able to turn off the use of weak cryptographic algorithms quickly when a vulnerability is discovered and limit use of cryptographic algorithms and/or adopt new cryptographic algorithms without making significant changes to infrastructures and without suffering from unnecessary disruptions.

Achieving crypto agility requires a systems approach to providing mechanisms that enable transition to alternate new algorithms and/or limit uses of existing algorithms in a seamless way while maintaining security and acceptable operation.

## 5. Comments/Suggestions on Section 3.3

**Section:** 3.3 Cryptographic Key Establishment – Page 11, Lines 453-454

**Comment:**

The listing of key establishment options in this sentence should be interpreted as including Hybrid Key Establishment as discussed at Line 431 and/or other future key establishment mechanisms.

It is suggested that lines 453-454 be revised to be more open-ended as follows:

Key establishment can includes— key-agreement mechanisms, key-transport mechanisms, and KEMs.

## **Comment to NIST.CSWP.39.ipd**

---

**Section:** 3.4 Balancing Security Strength and Protocol Complexity – Page 12, Lines 469

**Section:** 3.4.1 Balancing the Security Strength of Algorithms in a Cipher Suite – Page 12, Lines 473-475

**Comment:**

Neither the first sentence in Section 3.4 nor the second sentence in Section 3.4.1 provide a rationale for why “the relative strength of each algorithm should be roughly equal, or “[t]he algorithms in a cipher suite ought to provide roughly equal security strengths.””

In addition, the intended meaning of ‘the relative strength of each algorithm’ and ‘security strength’ in these sentences is not clear. For example, it is unclear whether and/or how the strength is measured or determined, and whether/how key length and Security Level might factor into such a determination.

In the case of a hybrid algorithm as described in Section 3.2.4 which uses both a traditional algorithm and a PQC algorithm, the relative strength of each algorithm is understandably different from a traditional public key algorithm that is more susceptible to attack by CRQCs than a PQC algorithm.

Beyond a clarification on the usage of ‘algorithm’ and ‘cipher,’ a definition of what is intended by the terminology ‘relative strength’ and ‘security strength’ is needed.

Even with such a definition of the intended meaning of ‘strength’ in these sections, there is still a need for an explanation of why these ‘strengths’ need to be roughly equal for each and every algorithm in a cipher suite of a security protocol. Such an explanation would allow for review and commentary by the stakeholders on the benefits and/or disadvantages of such a guideline.

## Comment to NIST.CSWP.39.ipd

---

### 6. Comments/Suggestions on Section 4.2

**Section:** 4.2 Using APIs in the Operating System Kernel – Page 14, Lines 548-560.

**Comment:**

This section does not mention embedded systems that might not have user and kernel spaces that are distinct from the processor's domains. Perhaps section between 4.2 and 4.3 could read something like the following:

Some security protocols in embedded systems must run within the real-time operating system (RTOS) kernel or privileged system tasks. An RTOS typically initializes early during device startup and manages critical hardware resources, task scheduling, and inter-process communication for the entire embedded application. For instance, secure communication protocols such as TLS offload encryption, authentication, and key management operations into system-level services within the RTOS to meet strict timing and security requirements.

To support cryptographic flexibility in this environment, the cryptographic API must be accessible to privileged code within the RTOS. However, in many embedded systems, only a minimal set of cryptographic primitives is included in the RTOS, selected based on the system's memory, real-time constraints, and security profile. In these cases, available algorithms are typically fixed at compile time, and dynamic loading of new cryptographic modules after deployment is often not supported due to reliability and certification concerns. Designers should consider the need to update the cryptographic implementation together or separate from the rest of the system.

Similarly, secure boot mechanisms and firmware authentication routines operate as part of the system startup sequence before the main application tasks begin. Some embedded platforms also include cryptographic self-tests during the boot process to validate the integrity and correct functioning of the cryptographic operations, ensuring that any malfunction or tampering is detected before normal system execution.

## **Comment to NIST.CSWP.39.ipd**

---

### **7. Public Comment about System-Level Security**

#### **Comment:**

While the draft provides valuable and detailed guidance on achieving crypto agility through algorithm selection, cipher suite negotiation, and protocol-level mechanisms, it tends to focus too narrowly on agility at the cryptographic primitive and protocol specification level. In doing so, it risks missing a broader and increasingly critical reality: that security is a system-level characteristic, and so too must be crypto agility.

To truly achieve crypto agility, the paper should more directly address the challenges and strategies for managing agility across entire systems, not just at the interfaces where cryptographic libraries or protocol suites are invoked. For example:

- The paper could expand its discussion on how to design system architectures that support dynamic substitution of cryptographic components without disrupting data flow, especially in tightly coupled environments such as embedded or constrained systems.
- There is limited treatment of how state synchronization, session continuity, backward compatibility and forward-migratability affect crypto agility across distributed systems or organizations. These operational realities present unique agility challenges not solvable by cryptographic design alone.
- The interaction between cryptographic agility and key lifecycle management, policy enforcement, and orchestration layers in large IT infrastructures also merits more detailed consideration.
- The discussion on hardware constraints rightly acknowledges fixed-function limitations, but the paper could go further in recommending system-level mitigations (e.g., layered encryption, flexible wrappers, runtime negotiation interfaces) to help balance performance, cost, and agility goals.

In short, the white paper would benefit from an explicit systems-thinking approach that treats crypto agility as a dynamic capability of the entire security architecture encompassing software, firmware, hardware, policies, and operational processes.