Cryptographic modules vs cryptographic algorithms

How algorithms, primitives and modules relate under FIPS 140-3, the CMVP and the CAVP, with short examples for AES, RSA and SHA-256.

Published August 30, 2025.

FIPS 140 | CRYPTOGRAPHY | CMVP

NOTE

Scope. The goal is to ground teams in how algorithms, primitives, and modules relate under **FIPS 140-3**, the **Cryptographic Module Validation Program (CMVP)**, and the **Cryptographic Algorithm Validation Program (CAVP)**. The paper is conceptual. It includes short, formal examples for AES, RSA, and SHA-256. No vendor-specific tooling.

Introduction

This whitepaper clarifies the distinction between **cryptographic algorithms** and **cryptographic modules**. It complements the companion whitepapers on CMVP and the Automata FIPS (AFIPS) model by focusing on theory and standards that drive real-world compliance in federal environments.

Definitions and Distinctions

Cryptographic Algorithms

A **cryptographic algorithm** is a mathematically defined procedure for security functions like encryption, decryption, hashing, digital signatures, message authentication, or key agreement. Examples include AES, RSA, ECDSA, SHA-256, and HKDF. Algorithms are specified in open standards and are independent of any specific product implementation.

Categories:

- **Symmetric ciphers**: one secret key for encryption and decryption, for example AES.
- **Asymmetric (public key)**: key pair for encryption or signatures, for example RSA and elliptic-curve schemes.
- **Hash functions**: one-way digests, for example SHA-256 and SHA-3.
- **MACs and AEADs**: integrity and authenticated encryption, for example HMAC and AES-GCM.

- **KDFs and KAS**: key derivation and key agreement, for example SP 800-56 and SP 800-56C methods.

Cryptographic Modules

A **cryptographic module** is a defined set of hardware, software, or firmware that implements security functions, protects plaintext keys, exposes controlled interfaces, and is contained within a **cryptographic boundary**. Validation of a module under **FIPS 140-3** assesses the module design, roles and services, key management, self-tests, physical protections, operational environment, and use of **approved** algorithms and key sizes.

Cryptographic Primitives vs Algorithms

A **cryptographic primitive** is a low-level algorithm used as a building block for higher-level schemes. Examples: a block cipher transformation, a stream cipher step, a hash compression function, an elliptic-curve scalar multiplication. A higher-level **algorithm** may be a primitive itself or a composition of primitives with additional rules. Examples: AES-CBC uses the AES block cipher plus CBC mode rules, RSA-PSS uses the RSA primitive plus hashing and padding, AES-GCM combines a block cipher with polynomial authentication.

- Security properties are defined at different levels. A primitive might be IND-CPA or collision resistant. A composed algorithm like AEAD targets confidentiality and integrity together.
- Validation and policy track both levels. The **approved list** names primitives and composite algorithms. Modules must implement and restrict usage accordingly.

References and Standards

- **FIPS 140-3**: Security Requirements for Cryptographic Modules - https://csrc.nist.gov/publications/detail/fips/140/3/final
- **CMVP**: Cryptographic Module Validation Program - https://csrc.nist.gov/projects/cryptographic-module-validation-program
- **CAVP**: Cryptographic Algorithm Validation Program - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program
- **SP 800-131A Rev. 2**: Transitioning the Use of Cryptographic Algorithms and Key Lengths - https://csrc.nist.gov/publications/detail/sp/800-131a/rev-2/final
- **SP 800-140C/140D**: Approved security functions and key establishment requirements - https://csrc.nist.gov/publications/detail/sp/800-140c/final and https://csrc.nist.gov/publications/detail/sp/800-140d/final
- **SP 800-38A/38D**: Block cipher modes of operation, including GCM - https://csrc.nist.gov/publications/detail/sp/800-38a/final and https://csrc.nist.gov/publications/detail/sp/800-38d/final
- **SP 800-56C Rev. 2**: Key-derivation methods - https://csrc.nist.gov/publications/detail/sp/800-56c/rev-2/final
- **FIPS 197**: Advanced Encryption Standard (AES) - https://csrc.nist.gov/publications/detail/fips/197/final
- **FIPS 180-4**: Secure Hash Standard (SHS) - https://csrc.nist.gov/publications/detail/fips/180/4/final
- **FIPS 186-5**: Digital Signature Standard - https://csrc.nist.gov/publications/detail/fips/186/5/final