CRYPTOGRAPHY POLICY

Version Number:	3.1
Version Date:	10/08/2025
Owned by:	Security, GRC
Approved by:	Nate Fitch, Sr. V.P., Chief of Staff
Confidentiality Level:	Internal

Table of Contents

1. Purpose

This policy establishes a formal framework for cryptographic controls throughout Docker's technology and data ecosystems to safeguard the confidentiality, integrity, authenticity, and non-repudiation of sensitive information including customer data, PII, intellectual property, and business-critical communications.

The policy mandates:

- Use of cryptographic algorithms vetted by recognized standards bodies (NIST, ISO, IETF)
- Prohibition of deprecated, insecure, or homegrown cryptographic implementations
- Centralized, automated systems for managing encryption keys and application secrets
- Approved processes for key generation, storage, rotation, revocation, and destruction
- Strong access controls, auditable workflows, and automated monitoring
- Comprehensive logging and regular review of cryptographic events

This framework ensures consistent cryptographic protection across Docker's cloud-native and hybrid environments, maintains user trust, fulfills regulatory and contractual obligations, supports end-to-end data security, and enables secure interoperability across systems and cloud providers.

2. Scope

This policy applies to all systems, platforms, personnel, and processes involved in the handling of sensitive information within Docker’s operational environment. It encompasses any service or application that stores, processes, or transmits data classified as Restricted, Confidential, or PII, whether in production or non-production environments.

The policy governs cryptographic protections across Docker-managed infrastructure, including internal corporate systems, public cloud environments, containerized workloads, CI/CD pipelines, and developer endpoints. It applies to all Docker employees, contractors, service providers, and external partners who design, deploy, or manage cryptographic functionality, or who handle credentials or encryption keys in any Docker environment.

Whether encryption is implemented for data in transit, data at rest, or data in use, the practices and controls described in this policy must be followed without exception.

3. Roles and Responsibilities

Role	**Responsibilities**
Security Team	Defines and maintains Docker’s cryptographic standards and policies. Reviews key and secret access logs, monitors for misuse or anomalies, responds to incidents, and leads investigations involving cryptographic systems.
DevOps / Infrastructure Team	Implements and maintains encryption tooling in cloud platforms and CI/CD pipelines. Ensures cryptographic configurations meet

	policy standards and enforces role-based access to cryptographic resources.
Engineering Team	Integrates encryption into applications and services, ensures secure transmission channels, and adheres to approved API and messaging security standards. Validates that sensitive data is appropriately encrypted in flight and at rest.
Compliance Team	Performs audits of cryptographic systems and key management processes. Supports documentation for SOC 2, ISO 27001, GDPR, and other frameworks requiring encryption evidence. Coordinates internal and external assessments.

4. Definitions

Key Management Service (KMS) - A cloud-native service that provides centralized control over cryptographic keys, enabling secure generation, storage, rotation, and access control while maintaining audit trails of all key operations.

Hardware Security Module (HSM) - A physical or virtual appliance that performs cryptographic operations and stores cryptographic keys in a hardened, tamper-resistant environment validated to FIPS 140-3 standards.

FIPS 140-3 - Federal Information Processing Standards Publication 140-3, a U.S. government standard that defines security requirements for cryptographic modules used to protect sensitive information.

Bring Your Own Key (BYOK) / Customer-Managed Key (CMK) - Cloud encryption models where customers maintain control over their encryption keys rather than using provider-managed keys, enabling greater control over data access and compliance with specific regulatory requirements.

Cryptographically Secure Pseudo-Random Number Generator (CSPRNG) - A random number generator that produces output indistinguishable from true randomness, essential for generating unpredictable encryption keys and other cryptographic material.

5. Cryptographic Use Cases

Cryptographic controls must be applied to a wide range of operational and engineering activities across Docker. Specific requirements include:

- Storage of customer data, PII, or sensitive internal records must be encrypted using AES-256 via cloud-native key management systems (KMS or HSM).
- Passwords must never be stored in plaintext and must be hashed using strong, adaptive algorithms such as PBKDF2, bcrypt, or Argon2 with unique, per-user salts.
- All web and API traffic must be secured using TLS, with TLS 1.3 preferred and TLS 1.2 as the minimum acceptable version.

6. Cryptographic Algorithm Standards

Only cryptographic algorithms validated by established standards bodies are permitted. Approved configurations include:

- Symmetric encryption must use AES-256 in either GCM or CBC mode, validated against FIPS 140-3 standards.
- Asymmetric encryption must use RSA with a minimum key size of 2048 bits (3072 preferred), or ECC with P-256 or stronger curves.
- Hashing algorithms must include SHA-256 or stronger variants such as SHA-3 or BLAKE2. MD5, SHA-1, and other deprecated hashes are not permitted.
- Digital signatures must use ECDSA or RSA-PSS. PKCS#1v1.5 signatures are considered insecure and are not approved for new implementations.
- Any use of insecure or deprecated protocols—including SSL, TLS versions below 1.2, RC4, DES, or homegrown ciphers—is explicitly prohibited.

7. Key Management Lifecycle

Docker applies strict controls throughout the cryptographic key lifecycle, aligned with industry standards recommendations:

- Keys must be generated using cryptographically secure pseudo-random number generators (CSPRNGs) and stored with unique identifiers.
- Key storage must take place in FIPS 140-3 validated KMS or HSM systems. Access to these systems must be tightly controlled using IAM roles and the principle of least privilege.
- Key distribution must occur only over encrypted channels with authenticated endpoints (e.g., mTLS). Private keys must never be embedded in code or exposed in repositories.
- Keys associated with sensitive data must be rotated at least every 90 days. Rotation should be automated via cloud-native tools (e.g., AWS/GCP KMS) where possible.
- Key revocation must occur immediately upon suspected compromise, job role change, or system decommissioning. Revocations must be logged and accompanied by an alert to the Security team.
- Destruction of cryptographic keys must follow industry standard media sanitization guidelines. All destruction actions must be timestamped, logged, and independently verified.
- All key usage must be logged, including the identity of the user or system, the operation performed, and the timestamp. Logs must be reviewed monthly and retained for a minimum of 12 months.

8. Secret Management

All application secrets must be managed using centralized secret management platforms approved by Docker. These include AWS Secrets Manager, AWS SSM Parameter Store, HashiCorp Vault, GCP Secret Manager, and 1Password.

- Secrets must be encrypted at rest and in transit using FIPS-validated cryptographic methods.
- Access to secrets must be governed by IAM policies enforcing least privilege, reviewed on a quarterly basis.
- Secrets must be rotated every 90 days or immediately upon detection of compromise. Secrets should also be assigned time-to-live (TTL) parameters and expiration alerts.

- Audit logs for secret access and modifications must be retained for at least 12 months and reviewed regularly for unauthorized access patterns.
- Hardcoding of secrets in source code, configuration files, CI/CD pipelines, or repositories is strictly prohibited.

9. Cloud Environment Requirements

To maintain consistent cryptographic control in public cloud environments:

- All libraries and modules used for encryption must be FIPS 140-3 validated or otherwise approved by Docker Security.
- All storage volumes and object stores must use default platform encryption (e.g., AWS EBS encryption, GCS bucket encryption).
- Data in transit must be protected by TLS 1.2, at a minimum, or equivalent secure protocol. Internal service-to-service communication must use mTLS where feasible.
- Customer environments must support Bring Your Own Key (BYOK) or Customer-Managed Key (CMK) models when required by contractual or regulatory requirements.
- Secrets and encryption keys must be accessible only to designated automation processes and explicitly authorized roles. Human access must be tightly restricted and monitored.

10. Enforcement and Exceptions

All Docker personnel are expected to comply with this policy in full. Failure to adhere may result in disciplinary action, up to and including termination of employment or contract. In cases involving willful misconduct, malicious activity, or violations of law, Docker may initiate legal proceedings. The organization reserves the right to monitor, audit, and inspect user activity on its systems to ensure compliance.

Any exceptions to this policy must be formally requested, documented, and approved by Docker's Legal & Compliance team. Exception requests must clearly outline the business justification and associated risks, and should be submitted to Docker's Risk Submission Portal.

11. Change History

Date	Version	Responsible Party	Description of Change
10/08/2025	3.1	Nate Fitch, Sr. V.P., Chief of Staff	Revised policy approved
09/25/2025	3.0	Chad Fryer, Sr. Security Engineer, GRC	Policy annual review and update
12/03/2024	2.2	Tushar Jain, Executive Vice President, Engineering	Revisions approval
5/20/2024	2.0	Karen Hajioannou, Senior Security & Compliance Engineer	Policy annual review and update

2/14/2023	1.2	Jean-Laurent de Morlhon, SVP of Engineering	Draft document approved
1/13/2023	1.1	Rachel Taylor, Senior Manager, Security, Risk & Trust	Draft Submitted for review
12/19/2022	1.0	Rachel Taylor, Senior Manager, Security, Risk & Trust	Basic document outline, contributions from Kat Yi