

# Simple Company Encryption Standard

Version 1.0 - Effective: January 15, 2026

Purpose: Establish a minimum, easy-to-follow encryption baseline for protecting company data in transit and at rest, including how keys are generated, stored, rotated, and audited.

## Minimum Requirements (Quick Reference)

Area	Standard (Minimum)
Data in transit	TLS 1.2+ (prefer TLS 1.3). Disable SSL, TLS 1.0/1.1.
Data at rest (databases, disks)	AES-256-GCM (or AES-256-CBC with HMAC-SHA-256 if GCM unavailable).
Passwords	Argon2id (preferred) or bcrypt. Never store plaintext or reversible encryption.
Keys and secrets	Use a KMS/Secrets Manager. No hard-coded secrets in code or images.
Key rotation	At least annually and upon suspected compromise; automate where possible.
Backups	Encrypted at rest with the same controls as production; tested restores.
Logging	Do not log sensitive plaintext (PII, secrets, tokens). Log key IDs and operation metadata only.

## 1. Scope

This standard applies to all company systems, applications, services, endpoints, and third-party vendors that store, process, transmit, or back up company data. It covers production, staging, development, and test environments (test data must be sanitized and treated as sensitive unless proven otherwise).

## 2. Data Classification (Simple)

**Public:** Approved for public release.

**Internal:** Non-public business data (default).

**Sensitive:** PII, authentication data, financial data, customer data, source code, or any regulated data.

**Restricted:** Secrets, encryption keys, credentials, private keys, high-risk regulated data, or data under legal hold.

Encryption is required for **Sensitive** and **Restricted** data in transit and at rest. Internal data should be encrypted where feasible and is required when stored outside managed corporate systems.

## 3. Encryption in Transit

All network communications carrying Sensitive or Restricted data must use TLS 1.2 or later (prefer TLS 1.3). This includes browser traffic, APIs, service-to-service calls, database connections, and administrative access.

**Minimum controls:**

- Disable SSL and TLS 1.0/1.1.
- Use strong cipher suites (AEAD preferred, e.g., AES-GCM or ChaCha20-Poly1305).
- Validate certificates; no "trust all" settings.
- Use HSTS for public web endpoints where applicable.
- Prefer mutual TLS (mTLS) for service-to-service in zero-trust or multi-tenant environments.

## 4. Encryption at Rest

Sensitive or Restricted data stored on disk must be encrypted using AES-256. Use authenticated encryption where possible.

**Accepted approaches:**

- Full-disk or volume encryption (e.g., cloud-managed disk encryption) for servers and workstations.
- Database encryption at rest (managed service features) plus field/column encryption for high-risk fields as needed.
- Object storage encryption (server-side, customer-managed keys preferred for Restricted data).

## 5. Approved Cryptography (Baseline)

Use case	Approved (minimum)	Notes
Symmetric encryption	AES-256-GCM	Prefer GCM; avoid ECB; ensure unique nonces/IVs.
Alt. symmetric (if GCM unavailable)	AES-256-CBC + HMAC-SHA-256	Encrypt-then-MAC; unique random IV.
Key exchange / public-key	ECDHE (P-256) or X25519	Prefer modern ECDHE for TLS.
Digital signatures	Ed25519 or ECDSA P-256	RSA allowed for compatibility only.
Legacy compatibility	RSA-3072+	Avoid new designs with RSA unless required.
Hashing / integrity	SHA-256 or SHA-384	No MD5/SHA-1 for security uses.
Password hashing	Argon2id or bcrypt	Use per-user salt; set work factors appropriately.

## 6. Key Management

Encryption is only as strong as key handling. All cryptographic keys and secrets must be generated, stored, and used through an approved key management system (KMS) or secrets manager.

**Rules:**

- Never hard-code secrets in source code, CI variables, container images, or documentation.
- Use envelope encryption: data keys (DEKs) encrypt data; key-encryption keys (KEKs) in KMS protect DEKs.
- Restrict key usage by role and service identity (least privilege).
- Separate environments: dev/stage/prod keys must be distinct.

- Maintain an inventory of keys with owners, purpose, creation date, and rotation cadence.

## 7. Rotation and Compromise

**Rotation:** Rotate KEKs at least annually and on personnel or vendor changes. Rotate DEKs based on data sensitivity and operational risk, and always after a suspected compromise. Prefer automated rotation features where available.

**Compromise response:** If a secret or key is suspected to be exposed, treat it as compromised: revoke access, rotate, invalidate dependent tokens, and investigate audit logs. For encrypted data, assess whether re-encryption is required based on exposure scope and key hierarchy.

## 8. Backups and Removable Media

Backups must be encrypted at rest and protected with the same (or stronger) access controls as production. Removable media is prohibited for Restricted data unless explicitly approved and encrypted with strong passphrases and managed keys.

Perform restore tests at least quarterly to ensure encryption does not prevent recovery and that key access procedures work.

## 9. Logging, Monitoring, and Telemetry

Do not log plaintext secrets, credentials, access tokens, private keys, or Sensitive/Restricted data fields unless formally approved for a specific incident and time-bounded. Logs should capture: key IDs (not key material), algorithm identifiers, service identity, timestamps, and success/failure codes.

## 10. Exceptions

Any deviation from this standard requires a documented exception with: business justification, compensating controls, scope, risk acceptance owner, and an expiration date. Exceptions must be reviewed at least every 6 months.

## Appendix A: Implementation Checklist

- Inventory systems that handle Sensitive/Restricted data.
- Confirm TLS 1.2+ everywhere; remove legacy protocols and weak cipher suites.
- Enable encryption at rest for databases, disks, and object storage.
- Centralize secrets in a secrets manager; remove secrets from repos and images.
- Create key ownership and rotation schedule; automate rotations when possible.
- Add CI checks to prevent committing secrets (secret scanning).
- Document incident playbooks for key compromise and rotation.
- Run quarterly restore tests for encrypted backups.