SOC 2 Control Narrative

Divy It Up, LLC

Document Creation Date:	March 15th, 2026
Effective Date:	March 15th, 2026
Version:	1.2
Classification:	Public

Area: Cryptographic Key Management & Sensitive Data Protection
Framework Alignment: SOC 2 Trust Services Criteria (Security, Confidentiality)

SOC 2 (System and Organization Controls 2) is an auditing procedure developed by the AICPA that ensures service providers manage customer data securely. It assesses controls—specifically security, availability, processing integrity, confidentiality, and privacy—to build trust, enhance reputation, and reduce data breach risks.

Control Objective
Sensitive financial data is protected using strong encryption and controlled key management processes such that:

- Confidential data is encrypted at rest and in transit
- Cryptographic keys are securely generated, stored, rotated, and destroyed
- Access to decryption capabilities is restricted, authenticated, and auditable

---

Control Design
CC6.1 / CC6.6 — Logical Access & Least Privilege
Access to cryptographic operations is restricted to a single backend service role with explicitly defined permissions:

- kms:GenerateDataKey
- kms:Decrypt
- kms:DescribeKey

No human users are granted direct cryptographic access to production keys.
Administrative users are limited to key lifecycle management and do not possess decrypt permissions.
This enforces **separation of duties** between:

- Key administration
- Cryptographic usage

CC6.7 — Credential Management
The system does not rely on long-lived credentials.

Instead:

- Access to AWS resources is obtained via **federated identity (OIDC)**
- Credentials are issued dynamically using **AWS STS**
- All credentials are **short-lived and automatically expire**

This eliminates:

- Static secrets in application code or configuration
- Credential rotation risk
- Exposure through repository or environment leakage

CC7.2 — Change Management (Cryptographic Configuration)

Changes to key policies, IAM roles, or cryptographic configuration require:

- Administrative access to AWS IAM/KMS
- Explicit modification of key policies or role permissions
- Audit logging via AWS CloudTrail

All changes are:

- Logged
- Attributable to a specific identity
- Reviewable post hoc

CC7.3 — Monitoring & Detection

All cryptographic operations are logged via **AWS CloudTrail**, including:

- GenerateDataKey
- Decrypt
- Role assumption events (AssumeRoleWithWebIdentity)

Logs include:

- Timestamp
- Calling principal
- Source IP / service context
- Encryption context metadata

These logs support:

- Anomaly detection
- Incident investigation
- Compliance evidence

CC8.1 — Encryption & Key Management

Sensitive data is protected using **envelope encryption**:

Key Structure

- Master keys: AWS KMS Customer-Managed Keys (HSM-backed)

- Data encryption keys: generated per record via KMS

Encryption Process

1. Application requests a data key from AWS KMS
2. Data is encrypted locally using AES-256-GCM
3. Encrypted data key and ciphertext are stored

Decryption Process

1. Encrypted data key is sent to AWS KMS
2. KMS returns plaintext data key
3. Data is decrypted in memory only

No plaintext sensitive data is stored at rest.

CC8.2 — Data Confidentiality

The following fields are encrypted at the application layer:

- Bank account numbers
- Routing numbers
- Third-party financial access tokens

Controls include:

- Immediate encryption upon ingestion
- No persistence of plaintext values
- Masking for display (e.g., last 4 digits only)

CC8.3 — Key Protection

AWS KMS provides:

- Hardware-backed key storage (HSM)
- Access control via key policies and IAM
- Automatic key rotation (where enabled)
- Separation of key material from application infrastructure

Application systems never have direct access to master key material.

CC9.2 — Data Retention & Minimization

The system enforces:

- Storage of only necessary sensitive data
- Avoidance of redundant duplication
- Separation of sensitive vs. non-sensitive fields

Where possible, tokenization or third-party custody is preferred over storage.

Control Effectiveness

The controls are effective because:

- Cryptographic keys are never exposed outside AWS KMS
- Access is limited to a single, scoped service role
- All usage is logged and auditable
- No plaintext sensitive data is persisted
- Credential compromise risk is minimized via federation