Encryption Policy

CredX Tech Inc

September 2026

Contents

Table 1: Control satisfaction

Standard	Controls Satisfied
TSC	CC9.9

Table 2: Document history

Date	Comment
Sep 4 2026	Confirmed storage-layer encryption at rest; app-layer gap named

1 Purpose and Scope

a. This policy defines organizational requirements for the use of cryptographic controls, as well as the requirements for cryptographic keys, in order to protect the confidentiality, integrity, authenticity and nonrepudiation of information.

b. This policy applies to all systems, equipment, facilities and information within the scope of CredX's information security program.

c. All employees, contractors, part-time and temporary workers, service providers, and those employed by others to perform work on behalf of CredX having to do with cryptographic systems, algorithms, or keying material are subject to this policy and must comply with it.

2 Background

a. This policy defines the high level objectives and implementation instructions for CredX's use of cryptographic algorithms and keys. It is vital that CredX adopt a standard approach to cryptographic controls across all work centers in order to ensure end-to-end security, while also promoting interoperability. This document defines the specific algorithms approved for use, requirements for key management and protection, and requirements for using cryptography in cloud environments.

3 Policy

4 Required Standards

a. CredX's Security Requirements Specification establishes the following as required standards:

   i. **TLS 1.2 or higher** for all data in transit. Connections using weak or deprecated cipher suites must be rejected. Mobile applications must implement certificate pinning for backend APIs.

   ii. **AES-256 or stronger** for all data at rest.

   iii. **Field-level (application-level) encryption** for high-risk PII attributes.

   iv. **Tokenization or pseudonymization** of identifiers wherever possible, and when sharing data with third parties.

b. **Key management requirements:**

   i. Encryption keys must be stored **separately from the encrypted data**.

ii. **Unique keys per environment** — development, staging, and production keys must not be shared.

iii. API keys, encryption keys, and signing secrets must be **rotated regularly**.

iv. Secrets must be stored only in Heroku Config Vars or another approved secrets manager, and **never committed to source control**.

v. During an incident, immediate key rotation and token revocation must be available as a containment action.

c. **Known exception — unencrypted credential storage.** The POS access and refresh tokens obtained during merchant POS connection are written to the database directly from the provider's OAuth response, and the wallet-pass rotating-code secret is likewise stored in the clear. **No encryption primitive of any kind exists anywhere in the backend** — directly inconsistent with the AES-256 encryption-at-rest requirement above.

i. By contrast, **merchant API keys are properly stored as salted hashes**, so this is a gap specific to POS integration credentials and the wallet-pass secret rather than a platform-wide absence of practice.

ii. **Practical impact**: until remediated, **database read access is effectively equivalent to POS access for connected merchants** — anyone with read access to the affected tables obtains a usable POS credential without defeating any additional encryption layer.

iii. **The exposure is not limited to database compromise.** The stored POS credential is exposed as an API field, and the single-merchant lookup enforces authentication but **not role authorization**, so any authenticated user of any role can read a merchant's live POS credential through the API. See the Access Onboarding and Termination Policy and the Security Architecture Narrative. Remediation should remove the field from the GraphQL model and add the role guard **before** encrypting at rest, since encryption does not close an authorized-read path.

iv. This is the **top-priority open remediation item for SOC 2 readiness**. *Owner: Ray Yip, CTO / Engineering. Target date: [TBD].*

d. **Storage-layer encryption at rest — confirmed.** Data at rest is encrypted at the platform layer in both stores: the **PostgreSQL** system of record (platform-managed AES-256) and **AWS S3 in `ca-central-1`**, which holds merchant logo images only. Encryption in transit over TLS 1.2+ is likewise confirmed.

e. **Application-layer field encryption — not implemented.** The requirement above for **field-level encryption of high-risk PII attributes**

has no implementation: no encryption primitive exists anywhere in the backend. All values are written to the database in the clear and depend entirely on platform disk encryption.

f. **These layers must not be conflated.** Storage-layer encryption protects against physical compromise of the underlying media. It does **not** protect against a database read, a compromised application credential, an over-permissive API path, or an insider with query access — in each of those the platform decrypts transparently and the reader sees plaintext. Confirmed at-rest encryption therefore does **not** close the plaintext credential exception above, and this policy must not be cited to that effect.

g. **Object-storage encryption is implicit rather than explicit.** Uploads rely on the storage bucket's default server-side encryption rather than requesting it explicitly. Objects are encrypted in practice, but setting it explicitly, or enforcing it through a bucket policy, would be materially stronger audit evidence. *Owner: Technical function.*

h. CredX must protect individual systems or information by means of cryptographic controls as defined in Table 3:

Name of System/ Type of Information	Cryptographic Tool	Encryption Algorithm	Key Size
Public Key Infrastructure for Authentication	OpenSSL	AES-256	256-bit key
Data Encryption Keys	OpenSSL	AES-256	256-bit key
Virtual Private Network (VPN) keys	OpenSSL and OpenVPN	AES-256	256-bit key
Website SSL Certificate	OpenSSL, CERT	AES-256	256-bit key

Table 3: Cryptographic Controls

b. Except where otherwise stated, keys must be managed by their owners.

c. Cryptographic keys must be protected against loss, change or destruction by applying appropriate access control mechanisms to prevent unauthorized use and backing up keys on a regular basis.

d. When required, customers of CredX's cloud-based software or platform offering must be able to obtain information regarding:

   i. The cryptographic tools used to protect their information.

   ii. Any capabilities that are available to allow cloud service customers to apply their own cryptographic solutions.

   iii. The identity of the countries where the cryptographic tools are used to store or transfer cloud service customers' data.

e. The use of organizationally-approved encryption must be governed in accordance with the laws of the country, region, or other regulating entity in which users perform their work. Encryption must not be used to violate any laws or regulations including import/export restrictions. The encryption used by the Company conforms to international standards and U.S. import/export requirements, and thus can be used across international boundaries for business purposes.

f. All key management must be performed using software that automatically manages access control, secure storage, backup and rotation of keys. Specifically:

g. The key management service must provide key access to specifically-designated users, with the ability to encrypt/decrypt information and generate data encryption keys.

h. The key management service must provide key administration access to specifically-designated users, with the ability to create, schedule delete, enable/disable rotation, and set usage policies for keys.

i. The key management service must store and backup keys for the entirety of their operational lifetime.

j. The key management service must rotate keys at least once every 12 months.

k. **Transport Layer Security (TLS):** All data in transit must be protected using TLS 1.2 or higher. TLS 1.0 and TLS 1.1 are deprecated and must not be used. Where technically feasible, TLS 1.3 is preferred. SSL in any version is prohibited.

l. **Certificate Lifecycle Management:**

m. All TLS/SSL certificates must be issued by a trusted Certificate Authority (CA).

n. Certificates must be inventoried and tracked, including expiry dates. An automated monitoring process must alert responsible parties at least 30 days before certificate expiry.

o. Expired certificates must be renewed before expiry. Any certificate found to have lapsed must be renewed immediately and the incident reviewed by the ISM.

p. Private keys associated with certificates must be stored securely and rotated upon suspected or confirmed compromise. Certificate revocation must be executed promptly using CRL or OCSP when a key is compromised.