Encryption & Key Management Policy

1. Purpose

This policy sets out how data is encrypted in transit and at rest, and how encryption keys and secrets are managed. Encryption protects data if it is intercepted on the network or if storage is accessed without authorisation; good key management makes sure that protection cannot be bypassed by obtaining the keys.

2. Scope

This policy applies to:

- All production data held by Subcontext, including customer data and backups
- All connections between users, the platform, its data stores and sub-processors
- All encryption keys, credentials, tokens and other secrets used by the platform
- Everyone who creates, holds or uses those keys and secrets

3. Definitions

Term	Meaning
Encryption in transit	Protecting data while it moves across a network, using TLS
Encryption at rest	Protecting data while it is stored, so it cannot be read without the key
Application-level encryption	An additional layer of encryption applied by the application to individual sensitive fields before they are stored
Hashing	A one-way transformation used to check a value without storing it in readable form
Secret	Any value that grants access, such as a password, API key, token or encryption key

4. Roles and Responsibilities

Role	Responsibility
Security Officer	Owns this policy, decides who may access keys and secrets, and oversees rotation
Engineers	Use the approved encryption mechanisms, never handle secrets outside the approved stores, and report any suspected exposure at once
Founding team	Holds access to production secrets needed for operation and recovery

5. Encryption in Transit

- All web traffic uses HTTPS, with TLS 1.2 or higher; plain HTTP is redirected to HTTPS
- Browsers are told to use HTTPS only (HTTP Strict Transport Security)
- Connections to the database require TLS
- Connections to the cache use TLS with an authentication token
- Connections to sub-processors use TLS

TLS 1.3 is supported where the client offers it. Content served through the content delivery network also requires TLS 1.2 or higher. Weak or outdated protocols are not accepted.

6. Encryption at Rest

Data	Protection
Database, including its backups	AES-256, managed by AWS
Cache	Encrypted at rest
File storage	AES-256 server-side encryption
Container images	Encrypted with AWS KMS
Integration credentials, connection secrets and webhook secrets	Additionally encrypted in the application (AES-256-GCM)
Settings, contact verification details and phone account tokens	Additionally encrypted in the application (AES-256-GCM)
Bank details for Direct Debit	Additionally encrypted in the application (AES-256-GCM)

Application-level encryption means that the most sensitive fields remain encrypted even to someone who could read the database directly. When a new type of sensitive field is added, the engineer decides with the Security Officer whether it needs application-level encryption, using the Data Classification Policy. Company devices used to access Subcontext systems use full disk encryption, as required by the Information Security Policy.

7. Passwords and Tokens

- User passwords are hashed with bcrypt, never stored in readable form
- API access tokens and application secrets are stored as hashes
- Publishable keys and browser approval tokens are looked up by a SHA-256 digest
- Card details are never stored by Subcontext: they are handled by Stripe

Because these values are hashed, they cannot be read back by Subcontext staff: a user who forgets a password resets it, and a lost API secret is replaced rather than recovered.

8. Key and Secret Management

- Database, cache and application master keys are held in AWS Secrets Manager

- Other secrets are held in the application's encrypted credentials, never in source code
- Access to keys and secrets is limited to the people and systems that need them
- Keys and secrets are rotated when someone with access leaves, and immediately on any suspected compromise
- Application encryption supports adding a new key while still reading data encrypted with the old one, so keys can be rotated without downtime

8.1 Rules for handling secrets

- Secrets must never be put in source code, tickets, chat messages, email or logs
- Secrets are not shared between environments: development and test never use production secrets
- Each secret has a clear purpose, and secrets that are no longer needed are revoked
- Secrets supplied by customers (for example credentials for their own systems) are stored with application-level encryption and used only to provide the service they were given for

8.2 Key lifecycle

Stage	Practice
Generation	Keys are generated by AWS or by the framework's secure random generator, never chosen by hand
Storage	Keys are held only in AWS Secrets Manager, AWS KMS or the application's encrypted credentials
Use	Keys are made available to running services at start-up and are not exposed in logs or error reports
Rotation	On a person with access leaving, on suspected compromise, and when the Security Officer judges it prudent
Retirement	Old application keys are kept only while data encrypted with them still exists, then removed

8.3 Worked example: suspected exposure

If a production secret were found in a log file, the person who found it would report it to the Security Officer at once. The secret would be rotated immediately, the log entry removed or restricted, and the audit and access logs checked for any use of the old value. The event would be handled under the Incident Response Plan, and the cause fixed so the secret could not be logged again.

9. Exceptions

Any departure from this policy, such as a connection to a supplier that cannot support TLS 1.2, must be approved in advance by the Security Officer and a director, recorded with its reason and compensating controls, and reviewed at least annually.

10. Monitoring Compliance

The Security Officer reviews at least annually who has access to keys and secrets, confirms that rotation has taken place when required, and checks that encryption settings in the infrastructure code match this policy.

11. Related Documents

- Information Security Policy
- Access Control Policy
- Data Classification Policy
- Secure Development Policy
- Incident Response Plan
- Staff Security Policy

12. Document Control

Item	Detail
Owner	Security Officer
Approved by	The directors of Subcontext Ltd
Review	At least annually
Effective	October 2026

Subcontext Ltd, registered in England and Wales, Company No. 17019645. Registered office: 124 City Road, London, EC1V 2NX. The current version of this document is published at subcontext.com.