

Cryptography and Key Management Policy

Encryption in transit and at rest, application-level encryption, hashing, and how keys are stored and rotated.

Version 1.0 · Last reviewed 22 October 2026 · Owner: Security Lead, security@etropo.com · Current version at <https://trust.etropo.com/resources/cryptography-policy>

1. Purpose

This policy states how Etropo uses encryption to protect data in transit and at rest, how it hashes credentials, and how it stores and rotates the keys involved.

2. Encryption in transit

- Every connection to app.etropo.com and www.etropo.com uses HTTPS. Certificates are issued and renewed automatically by Vercel.
- HTTP Strict Transport Security is set for two years and covers every subdomain, so browsers refuse plain HTTP.
- Connections from the application to the database, to Supabase Auth, and to every third-party API use HTTPS or TLS-protected protocols only.
- Supported TLS versions and cipher suites are managed by Vercel and follow current browser standards; obsolete versions are not offered.

3. Encryption at rest

- The PostgreSQL database, file storage and backups are encrypted at rest by Supabase on AWS with AES-256.
- Vercel environment variables are encrypted at rest and decrypted only into the running function.

4. Application-level encryption

OAuth access and refresh tokens for customer-connected services (Google, LinkedIn, Meta, TikTok, Notion, HubSpot, Attio, Slack) are encrypted before they are written to the database:

- algorithm AES-256-GCM;
- a random 128-bit initialisation vector per record;
- the GCM authentication tag stored with the ciphertext, so a tampered value fails to decrypt;
- the key is a 256-bit value held only in the production environment's encrypted settings.

A database read without the key yields nothing usable.

5. Hashing and secrets

- Passwords are hashed by Supabase Auth with bcrypt. Etropo code never handles a plain-text password.
- Two-factor secrets are generated and verified by Supabase Auth; the TOTP secret is shown once at enrolment and not stored by Etropo code.
- OAuth state values and other tokens are generated with the platform's cryptographically secure random generator.
- Webhook payloads are verified with the signing secret of each sender (HMAC for Stripe, Slack, Resend and Supabase; Bot Framework tokens for Microsoft Teams).

6. Key management

Key or secret	Where it lives	Who can read it	Rotation
Token-encryption key	Vercel production environment	Security Lead and Infrastructure Lead	On suspected exposure; otherwise reviewed annually
Supabase service-role key	Vercel production environment	Same	On suspected exposure; otherwise reviewed annually
Vendor API keys and OAuth client secrets	Vercel environment, per environment	Same	On suspected exposure; on vendor advice
Webhook signing secrets	Vercel environment	Same	When the webhook endpoint is recreated
TLS private keys	Vercel	Vercel	Automatic
Database encryption keys	AWS KMS via Supabase	Supabase	Managed by the provider

Rules:

- Production and preview environments use different values for every secret.
- Secrets are never committed to Git, pasted into chat, or stored on personal devices.

- A compromised key is rotated immediately and the Security Incident Response Plan is followed.
- Rotating the token-encryption key requires re-encrypting every stored token. A tested rotation procedure is on the readiness plan.

7. Review

Reviewed annually. Version 1.0, October 2026.

This PDF is a snapshot. The page at <https://trust.etropo.com/resources/cryptography-policy> is the current version.