Data Encryption Policy

Data Encryption Policy

> **Project / Organization:** Bilko — Balkan Accounting SaaS **Policy Number:** POL-SEC-ENC-001 **Version:** 1.0 **Date:** 2026-02-23 **Author:** Compliance Architect **Status:** Draft **Reviewers:** CTO, DPO, Engineering Lead **Next Review:** 2026-08-23 **Classification:** Confidential

Document History

Version	Date	Author	Changes
0.1	2026-02-23	Compliance Architect	Initial draft — Bilko encryption standards

---

1. Purpose & Scope

Purpose: This policy defines minimum encryption standards for all data at rest, in transit, and at the application level across all systems operated by Bilko.

Scope:

- All systems operated by Bilko (api.bilko.io, bilko.io, PostgreSQL, Cloudflare R2)
- All employees, contractors, and third parties with access to Bilko systems
- All data classified Internal, Confidential, or Restricted (see compliance-framework.md §6)
- Railway PostgreSQL (EU West), Vercel (frontend), Cloudflare R2 (file storage)

Regulatory basis:

- GDPR Art. 32 — Appropriate technical measures including encryption
- ZZPL Art. 50 (Serbia) — Security of personal data processing
- ZZLP Art. 14 (BiH) — Technical data protection measures
- Zakon o računovodstvu (RS/HR/BA) — Integrity of financial records

2. Encryption Standards & Approved Algorithms

2.1 Approved Algorithms

Symmetric Encryption

Use Case	Algorithm	Key Size	Mode	Notes
Data at rest (general)	AES	256-bit	GCM	Authenticated encryption — default
Database disk encryption	AES	256-bit	XTS	Railway PostgreSQL default
File storage	AES	256-bit	GCM	Cloudflare R2 server-side
Backup encryption	AES	256-bit	GCM	Railway automatic backup encryption

Asymmetric Encryption

Use Case	Algorithm	Key Size	Notes
TLS key exchange	ECDHE	P-256 / P-384	Cloudflare + Railway TLS 1.3
JWT signing	HMAC-SHA-256 (HS256)	256-bit	JWT_SECRET (32+ chars, CSPRNG)
JWT refresh	HMAC-SHA-256 (HS256)	256-bit	JWT_REFRESH_SECRET (separate key)
Future: JWT asymmetric	Ed25519 (EdDSA)	256-bit	Planned Phase 2 migration

Hashing & Password Storage

Use Case	Algorithm	Parameters	Notes
Password hashing	bcrypt	cost factor = 12	`bcrypt.hash(password, 12)`
Token hashing (refresh tokens)	HMAC-SHA-256	—	Refresh tokens hashed before DB storage
Data integrity (general)	SHA-256	—	File checksums, non-security hashing

2.2 Prohibited Algorithms (NEVER USE)

Algorithm	Reason
MD5	Collision attacks (2004+) — completely broken
SHA-1	Collision attacks (2017+)
DES / 3DES	Key size insufficient
RC4	Statistical biases
ECB mode (any cipher)	Leaks data patterns — deterministic
RSA < 2048-bit	Insufficient key strength
AES-128 for Restricted data	Insufficient for financial/tax ID data
bcrypt < 12 rounds	Insufficient work factor
MD5 for password hashing	NEVER — use bcrypt

---

3. Encryption at Rest

3.1 Database Encryption

Database	Method	Key Management	Coverage
PostgreSQL (Railway EU West)	AES-256 disk encryption (Railway TDE)	Railway-managed	All data classifications
PostgreSQL — tax IDs (PIB/JMBG/OIB/JIB)	AES-256-GCM application-layer field encryption	Environment variable (Railway secrets)	L4 Restricted
PostgreSQL — IBAN	AES-256-GCM application-layer field encryption	Environment variable (Railway secrets)	L4 Restricted
PostgreSQL backups	AES-256 (Railway automatic)	Railway-managed	All data

Field-level encryption for L4 Restricted data:

```
// Tax IDs and bank account numbers encrypted at application layer
// In addition to Railway disk encryption
import crypto from 'crypto';

async function encryptRestrictedField(plaintext: string): Promise<string> {
  const key = Buffer.from(process.env.FIELD_ENCRYPTION_KEY!, 'hex'); // 32 bytes
```

```
  const iv = crypto.randomBytes(12); // 96-bit IV for GCM
  const cipher = crypto.createCipheriv('aes-256-gcm', key, iv);
  const ciphertext = Buffer.concat([cipher.update(plaintext, 'utf8'), cipher.final()]);
  const tag = cipher.getAuthTag();
  // Store: base64(iv || tag || ciphertext)
  return Buffer.concat([iv, tag, ciphertext]).toString('base64');
}

async function decryptRestrictedField(stored: string): Promise<string> {
  const key = Buffer.from(process.env.FIELD_ENCRYPTION_KEY!, 'hex');
  const buf = Buffer.from(stored, 'base64');
  const iv = buf.subarray(0, 12);
  const tag = buf.subarray(12, 28);
  const ciphertext = buf.subarray(28);
  const decipher = crypto.createDecipheriv('aes-256-gcm', key, iv);
  decipher.setAuthTag(tag);
  return decipher.update(ciphertext) + decipher.final('utf8');
}
```

3.2 File Storage Encryption

Storage	Method	Key Management	Notes
Cloudflare R2 (receipts, invoice PDFs)	AES-256 server-side (Cloudflare default)	Cloudflare-managed	EU region bucket required

3.3 Backup Encryption

Railway provides automatic daily PostgreSQL backups, encrypted with AES-256. Backup retention: 30 days (Railway default). Custom backup strategy to be implemented in Phase 2 with separate backup encryption key stored in Railway environment secrets.

---

4. Encryption in Transit

4.1 TLS Configuration

Minimum TLS version:

- External-facing (api.bilko.io, bilko.io): TLS 1.3 (enforced via Cloudflare)

- Railway internal: TLS 1.3

Approved cipher suites (TLS 1.3):

```
TLS_AES_256_GCM_SHA384
TLS_CHACHA20_POLY1305_SHA256
TLS_AES_128_GCM_SHA256 (minimum)
```

HSTS configuration:

```
Strict-Transport-Security: max-age=63072000; includeSubDomains; preload
```

HTTP redirect: All HTTP traffic redirected to HTTPS (301). No HTTP allowed in production.

Prohibited:

- TLS 1.0 and TLS 1.1 — prohibited
- SSL 2.0 / SSL 3.0 — prohibited
- RC4 cipher suites — prohibited

4.2 Cookie Security

All session cookies set with:

```
res.cookie('refreshToken', token, {
  httpOnly: true,    // Not accessible to JavaScript
  secure: true,      // HTTPS only
  sameSite: 'strict', // CSRF protection
  maxAge: 7 * 24 * 60 * 60 * 1000, // 7 days
});
```

4.3 API Security Headers (Helmet.js)

```
app.use(helmet({
  hsts: { maxAge: 63072000, includeSubDomains: true, preload: true },
  contentSecurityPolicy: {
    directives: {
      defaultSrc: ["'self'"],
      scriptSrc: ["'self'", "'unsafe-inline'"],
      styleSrc: ["'self'", "'unsafe-inline'"],
      imgSrc: ["'self'", "data:", "https:"],
```

```
    },
  },
}));
```

---

5. Application-Level Encryption

5.1 Field-Level Encryption Requirements

Required for all L4 Restricted fields:

Field	Table	Reason
`taxId` (organization tax ID: PIB/JIB/OIB)	`organizations`	National business identifier
`taxId` (contact tax ID: PIB/JMBG/OIB/JIB)	`contacts`	National person/business identifier
`iban`	`organizations`, `contacts`, `bankAccounts`	Bank account number
`totpSecret`	`users`	2FA seed — must not be readable

5.2 JWT Token Encryption

JWTs signed with HMAC-SHA-256 (HS256) using `JWT_SECRET`:

- `JWT_SECRET`: 32+ character random string (CSPRNG), stored in Railway secrets
- `JWT_REFRESH_SECRET`: Separate 32+ character key — never same as JWT_SECRET
- Access token payload: `{ sub: userId, org: organizationId, role, iat, exp }`
- NO PII in JWT payload — no email, name, tax ID

Refresh tokens stored as HMAC-SHA-256 hash in database — raw token never stored.

5.3 Password Hashing

```
import bcrypt from 'bcrypt';

// Registration
const hash = await bcrypt.hash(plainPassword, 12);

// Login verification
```

```
const isValid = await bcrypt.compare(plainPassword, storedHash);
```

bcrypt parameters: cost factor 12. Never downgrade below 12.

---

6. Key Management Summary

Full policy: key-management-policy.md

Key Type	Storage	Rotation	Owner
`JWT_SECRET`	Railway environment secret	Quarterly	Security
`JWT_REFRESH_SECRET`	Railway environment secret	Quarterly	Security
`FIELD_ENCRYPTION_KEY` (tax IDs, IBAN)	Railway environment secret	Annual	Security
PostgreSQL disk encryption	Railway-managed (TDE)	Railway-managed	Railway
Cloudflare R2 encryption	Cloudflare-managed	Cloudflare-managed	Cloudflare
TLS certificates (Cloudflare)	Cloudflare Certificate Manager	90 days (automatic)	Cloudflare

Secrets never committed to git. `.env` files in `.gitignore`. All secrets managed via Railway environment variables (production) or `.env.local` (development, git-ignored).

---

7. Cryptographic Inventory

System	Algorithm	Key Size	Mode	Key Location	Next Rotation
PostgreSQL disk (Railway)	AES-256	256-bit	XTS	Railway-managed	Railway-managed
Tax ID field encryption	AES-256-GCM	256-bit	GCM	Railway env secret	Annual
IBAN field encryption	AES-256-GCM	256-bit	GCM	Railway env secret	Annual
Cloudflare R2	AES-256	256-bit	—	Cloudflare-managed	Cloudflare-managed
External TLS (Cloudflare)	ECDSA P-256	256-bit	—	Cloudflare	90 days (auto)
JWT signing	HMAC-SHA-256	256-bit	—	Railway env secret	Quarterly

System	Algorithm	Key Size	Mode	Key Location	Next Rotation
Refresh token signing	HMAC-SHA-256	256-bit	—	Railway env secret	Quarterly
Password hashing	bcrypt	cost=12	—	N/A	N/A

---

8. Financial Data Integrity

Financial data requires not just confidentiality but also integrity. Bilko enforces:

1. **NUMERIC(19,4) for all monetary amounts** — never float or JavaScript number. Prevents rounding errors in VAT calculations.
2. **Immutable LoggedAction table** — all mutations append-only with old/new values. Enables financial audit.
3. **Double-entry enforcement** — debit = credit validated at backend. Prevents imbalanced entries.
4. **Exchange rate locking** — rates stored at transaction date. Historical accuracy preserved.

---

9. Exception Process

Exceptions not permitted for: L4 Restricted data (tax IDs, IBAN, TOTP secrets, password hashes) — no exceptions.

Exception request process:

1. Submit to: security@bilko.io
2. Required: system affected, algorithm excepted from, business justification, risk assessment, compensating controls, proposed duration (max 12 months)
3. Approval: CTO
4. Log: `/docs/security/exceptions.md`
5. Review: Quarterly

Active exceptions: None at this time.

Approval

Role	Name	Date	Signature

Author	Compliance Architect	2026-02-23	
CTO			
DPO			
Engineering Lead			

Revision #15
Created 2026-02-23 12:02:59 UTC by John
Updated 2026-06-28 20:02:58 UTC by John