SQL Database
Encryption Practices

- Encryption as a critical defense-in-depth control
- Scope: at rest, in transit, in use, key management
- Outcomes for DBAs, backend engineers, security practitioners
- Covers threats, compliance, implementation, and key handling

Why Encrypt Databases

- - Threats: stolen media, backups, sniffing, malicious insiders
- - Compliance drivers: HIPAA, PCI DSS, GDPR, SOC 2, CMMC
- - Encryption does not replace access control or patching

Encryption at Rest: Threat Model and Compliance Drivers

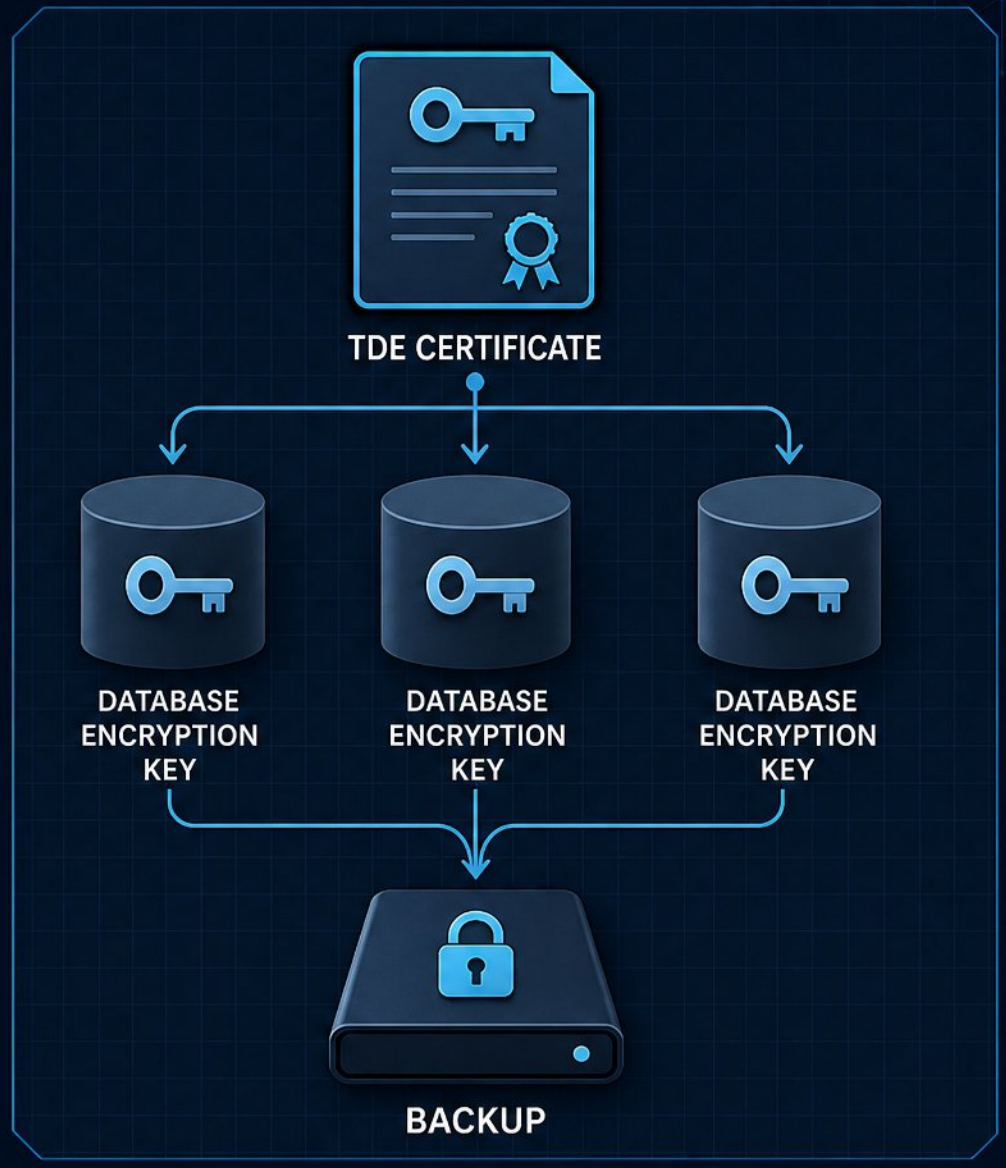
- At-rest encryption guards stolen disks, backups, and media
- Does not replace access control, patching, or query security
- HIPAA safe harbor, PCI DSS scope, and CMMC FIPS controls
- Must secure backups, snapshots, and replicas, not just data

Encryption at Rest: Technologies and Options

- TDE encrypts data files transparently; zero app changes
- File/volume encryption protects OS, logs, and temp files
- SQL Server, Oracle offer native TDE with automatic key hierarchy
- MySQL, PostgreSQL rely on volume or application-layer encryption
- Consider performance overhead, recovery needs, and key hierarchy

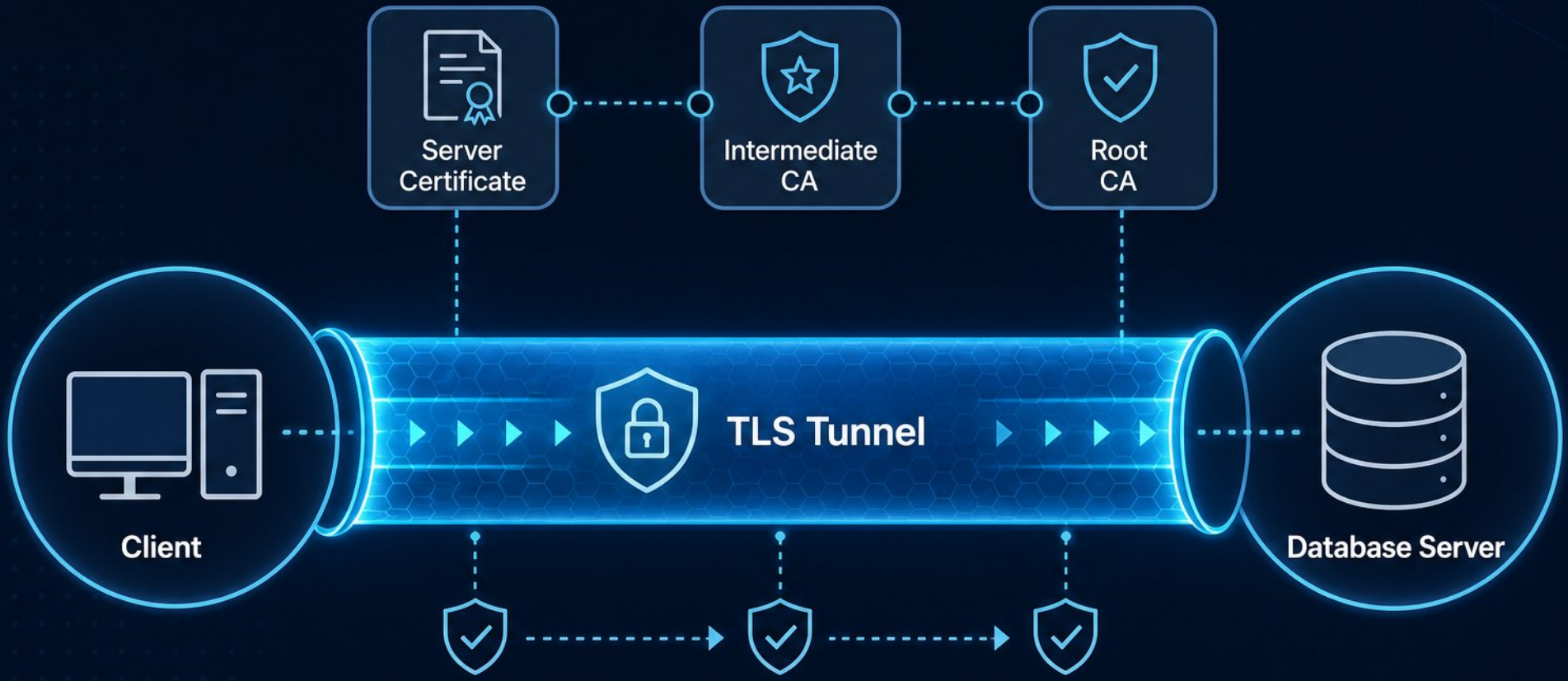
Encryption at Rest: Implementation Best Practices

- Back up TDE certificates and keys immediately after creation
- Store database encryption keys separately from the database
- Never share wallets across environments or teams
- Use a dedicated key management system, not ad-hoc keystores
- Encrypt all backups, exports, and replication traffic



Encryption in Transit: TLS Configuration

- TLS secures client-server connections and replication streams
- Validate certificates against trusted root CAs
- Enforce TLS 1.2+ with strong cipher suites
- Detect expired certificates, disabled verification, or weak protocols



Encryption in Transit: mTLS and Platform Patterns

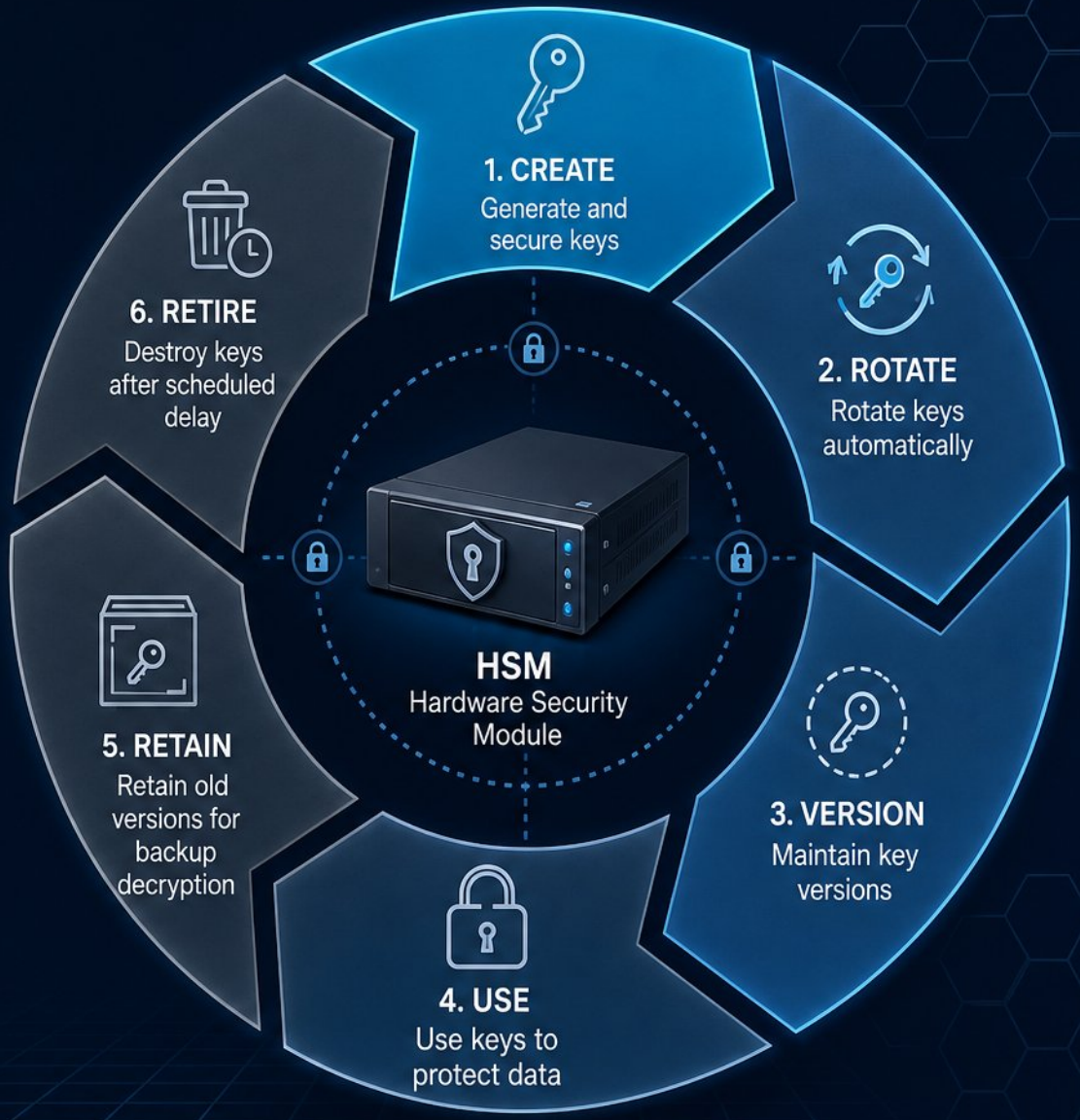
- Mutual TLS verifies both client and server identities
- Platform-specific setup for PostgreSQL, MySQL, SQL Server, Oracle, MongoDB
- Detect weak ciphers, expired certs, and disabled verification
- Prevent downgrade and man-in-the-middle attacks
- Automate certificate rotation and enforce TLS 1.2+

Column-Level and Application-Layer Encryption

- Column-level encryption for selective sensitive data
- TDE protects whole database; columns add finer control
- Envelope encryption: data key wrapped by master key
- Application-managed keys limit DBA access
- Encrypted columns hinder search, indexes, equality queries

Key Management: Lifecycle and Architecture

- Centralize keys per environment, separate from encrypted resources
- Rotate keys automatically, ideally at least once yearly
- Use HSM-backed key protection for high-sensitivity data
- Assign distinct roles: key admins vs. key users
- Grant least-privilege permissions; audit all key operations via CloudTrail/logging
- Destroy keys only after scheduled delay; retain old versions for backup decryption

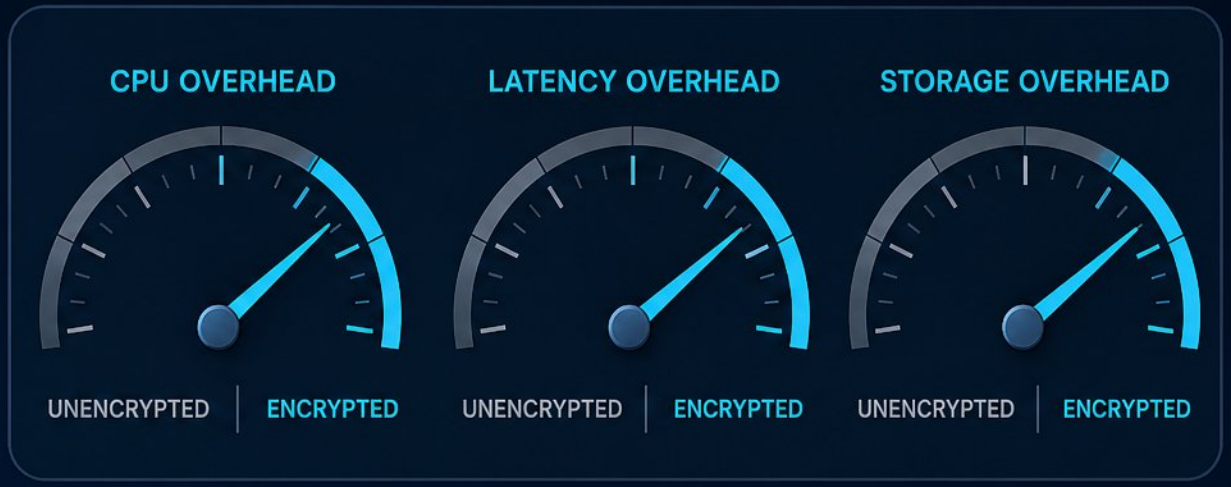


Key Management: Cloud KMS and Emergency Scenarios

- Match KMS, HSM, or BYOK to compliance and control needs
- Use cloud-generated key material and customer-managed keys
- Enforce separation of duties between key admins and key users
- Disabling a key can render encrypted databases inaccessible
- Rotate on schedule; document key-revocation and recovery playbooks

Performance and Operational Impact

- - TDE overhead under 10% for typical OLTP workloads; 30%+ for bulk rewrites
- - WAL encryption worst cases around 20%, improved in pg_tde 2.2.3
- - Low concurrency overhead higher (up to 20%); high concurrency below 5%
- - Capacity planning: add CPU headroom for AES-NI encryption operations
- - Test backup, restore, and DR performance with encryption enabled



Testing and Auditing Encryption Controls

- Verify data encryption at rest and in transit
- Automate encryption checks in CI/CD pipelines
- Use infrastructure scanning for continuous validation
- Collect clear evidence for compliance audits
- Test key management and rotation processes

Common Pitfalls and Real-World Incidents

- Hardcoded keys and secrets in public repos (“Toyota, 2023”)
- Plaintext backups or S3 buckets exposed publicly (CVS, Estée Lauder)
- Expired TLS inspection certificate blinded Equifax defenders for 10 months
- 82% of S3 misconfigs trace to human error—no automated guardrails
- Encrypted volumes do not prevent credential theft or permissive RLS policies

Practical Implementation Roadmap

- Classify database risk via threat modeling
- Phase rollout: inventory, pilot, key management, deployment
- Centralize key management and separation of duties
- Validate encryption controls through automated tests
- Use checklists for DBAs, engineers, and security teams

PersonWise.

PersonWise • AI digital-human course platform

Scan the code or click anywhere to watch this course live, with voice Q&A

personwise.ai/t/8c081fa5/public