Mydbops
Scaling Databases

Securing MySQL

Data at Rest Encryption

Mydbops MyWebinar 50

Presenter

Praveen GR

Organization

Mydbops

Mydbops by the Numbers

MySQL

9+ years

Of Expertise

10 B +

DB Transactions Handled per Day

800 +

Happy Clients

3000 +

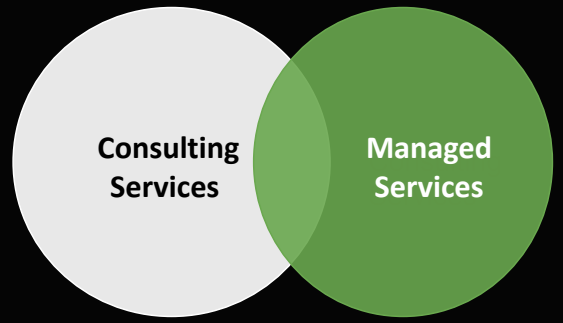
Tickets Handled per Day

Your Trusted Database Management Partner

With 9+ Years of Expertise

Mydbops
Scaling Databases

Our Services



Consulting Services

Targeted Engagement

Managed Services

24×7 DBA Team

Database Technologies

MySQL · MongoDB · PostgreSQL · MariaDB · TiDB · Cassandra

Mydbops
Scaling Databases

Real-World Threat Model: What Encryption Actually Protects

Attacker Copies .ibd Files

Direct file system access to tablespace data

Stolen Physical Disk

Hardware theft from data center or storage

Compromised Backup Repository

Access to backup storage or archives

Host-Level Breach

Operating system compromise access

Encryption protects data files — not active MySQL sessions

Internal Encryption Model in MySQL

InnoDB Tablespace Encryption

Protects data at the storage level

Page-Level Encryption

Granular data protection mechanism

Transparent to Application

No application code changes required

Keyring-Managed Master Keys

Secure key storage and rotation

Encryption/Decryption Inside Buffer Pool

All cryptographic operations occur in memory, ensuring data never exposed in plaintext on disk

Encryption Architecture

MySQL

1

Master Key (In Keyring)

Root key stored securely in keyring plugin

2

Tablespace Key (Per Tablespace)

Unique encryption key for each tablespace

3

Key Encryption (Dual-Layer)

Tablespace key encrypted with master key

4

Memory-Only Decryption

Plaintext keys never written to disk

5

AES-Based Encryption (Industry Standard)

Advanced Encryption Standard for robust security

Mydbops
Scaling Databases

Encryptable Components

File-per-table Tablespaces

Individual .ibd files for each table

General Tablespaces

Shared tablespace containers

Undo Tablespaces

Rollback segments storage

Redo Logs

Write-ahead transaction logs

Binary Logs

Replication event streams

Relay Logs

Replica server log files

Why Each Encrypted Layer Matters

MySQL

Tablespaces

Primary Target

Actual business data

- Contains all user data
- Most critical asset
- Direct breach impact

Undo Tablespaces

Hidden Risk

Old row versions

- Historical data versions
- Deleted sensitive data
- Rollback information

Redo Logs

Transaction Trail

Recent changes

- Transaction history
- Sensitive row images
- Before/after values

Mydbops
Scaling Databases

Why Each Encrypted Layer Matters

Binary Logs

Replication Stream

Full SQL/row-based events

- Contains all database changes
- Critical for point-in-time recovery
- Replication topology dependency

Relay Logs

Replicated Data

Sensitive data on replicas

- Mirror of master's binary logs
- Same sensitive content exposure
- Required for async replication

Why Data-at-Rest Encryption is Non-Negotiable

MySQL

Disk Theft / VM Snapshot

Physical storage compromise or virtual machine snapshot exposure

Backup File Exposure

Unprotected backup repositories containing sensitive data

Storage Breach

Direct access to storage systems and data files

Compliance Mandates

PCI-DSS, GDPR, HIPAA regulatory requirements

Insider File-Level Access

Unauthorized file system access by privileged users

Mydbops
Scaling Databases

Key Management Architecture

Master Key Storage

Securely stored in keyring plugin

Per-Tablespace Keys

Unique encryption keys for each tablespace

Key Hierarchy Model

Layered encryption structure for security

Online Rotation

Master key rotation without downtime

Critical Availability

Keyring availability = data availability

MySQL 8.0 Implementation: Keyring Plugin Model

Configuration in my.cnf

```
early-plugin-load keyring_file.so
```

```
keyring_file_data /path/to/keyring
```

Verification Command

```
SHOW PLUGINS;
```

Confirm keyring_file plugin is active

Important Notes

- Plugin must load before InnoDB initializes
- Secure keyring file location with restricted permissions
- Backup keyring file regularly

Mydbops
Scaling Databases

MySQL 8.4 Implementation: Component-Based Keyring

Component Infrastructure

component_keyring_file

Keyring component file : component_keyring_datafile

Actual key storage file

Keyring component cnf : component_keyring_file.cnf

Path to the keyring data file

Keyring component manifest file : mysqld.my

The keyring must load before InnoDB initializes.

MySQL 8.4 Implementation: Component-Based Keyring

Verification Query

```
-- Check component status

SELECT * FROM

performance_schema.keyring_component_status;
```

Verify component_keyring_file is active

Key Advantages

- Modern architecture with better flexibility
- Improved security and maintainability
- Easier upgrade path for future versions

8.4 vs 8.0

8.0 Plugin	8.4 Component
early-plugin-load	**No preload needed**

Mydbops
Scaling Databases

Tablespace Encryption Strategy

At Creation

```
-- New table with encryption

CREATE TABLE table_name ( ... )
ENCRYPTION = 'Y';
```

Enable encryption when creating new tables

Existing Tables

```
-- Encrypt existing table

ALTER TABLE table_name
ENCRYPTION = 'Y';
```

Encrypt already existing tables on demand

Instance Default

```
-- Global default setting

default_table_encryption = ON;
```

Automatically encrypt all new tables

Mydbops
Scaling Databases

Redo & Undo Log Encryption Controls

Redo Log Encryption

```
-- my.cnf configuration
innodb_redo_log_encrypt = ON
```

Undo Log Encryption

```
-- my.cnf configuration
innodb_undo_log_encrypt = ON
```

Instance-Wide Impact

These settings affect the **entire MySQL instance**

- All databases and tables
- Requires careful planning
- Test in staging environment first

Binary & Relay Log Encryption

Configuration

```
-- my.cnf configuration
binlog_encryption = ON
```

Binary Logs

All database changes recorded

Key Points

- Automatically encrypts relay logs
- Protects replication stream data
- Enabled for entire instance

Relay Logs

Replicated events on standby servers

Master Key Rotation & Lifecycle Management

Rotation Command

```
-- Rotate master key online
ALTER INSTANCE ROTATE INNODB MASTER KEY;
```

Online Operation

No downtime required. Rotation happens in the background while the database remains fully operational.

When to Rotate

- Compliance requirements (PCI, HIPAA)
- Scheduled security policies
- After security incidents

Re-encrypts Tablespace Keys

Generates new master key and re-encrypts all tablespace encryption keys with the new master key.

Compliance Control

Backup & Recovery Implications

Physical Backup Requires Keyring

Keyring file must be backed up alongside database files

Keyring Loss = Data Unrecoverable

Without keyring, encrypted data cannot be decrypted

Logical Dump Not Encrypted

mysqldump exports plaintext data

Secure Keyring Storage

Mandatory for recovery

DR Testing Essential

Validate recovery process

Standard Security Practices

MySQL

Restrict OS-Level Access

Limit operating system access to database server and keyring file location

Keyring File Permissions

Set restrictive file permissions (600) owned by mysql user only

Separate Keyring Storage

Store keyring file on separate encrypted filesystem or hardware security module

Key Rotation Schedule

Implement regular master key rotation following compliance requirements

Minimal Privileges

Grant least necessary privileges to database users and encryption management accounts

Mydbops
Scaling Databases

Operational Considerations

Dynamic Log Encryption

MySQL Service restart is not required.

- Redo log encryption
- Undo log encryption
- Binary log encryption

Replication Topology Alignment

Ensure consistent encryption across:

- Primary and replica servers
- Same keyring configuration
- Compatible MySQL versions

Rolling Enablement Strategy

Gradual deployment approach:

- Test environment validation
- Start with non-critical tables
- Monitor performance impact

Monitor Encryption Metadata

Track encryption status:

- Tablespace encryption status
- Key rotation history
- Performance metrics

Performance Impact

MySQL

CPU Overhead

AES encryption adds computational cost to each data operation

Additional I/O

Extra disk operations for encryption/decryption processes

Write Workloads

Heavy write operations → Higher performance cost

Benchmark Before Rollout

Test encryption performance impact in staging environment matching production workload patterns before production deployment

Commit Latency

Increased transaction commit time due to encryption overhead

Mydbops
Scaling Databases

Performance Impact

MySQL

Thread Activity

Monitor thread contention and concurrency patterns

Average Query Latency

Before	After
12.3 ms	14.7 ms

Commit Latency

Before	After
1.2 ms	1.8 ms

CPU Usage

Higher utilization during encryption operations

P95 Latency

Before	After
45 ms	52 ms

P99 Latency

Before	After
89 ms	108 ms

IOPS

Additional disk I/O for encryption/decryption

Application Latency

End-to-end response time impact analysis

Mydbops
Scaling Databases

Phased Rollout Strategy for Production

1 Phase 1

Configure Keyring

Install and configure keyring plugin or component

Encrypt New Tables

Enable default_table_encryption=ON

Foundation Setup

2 Phase 2

Encrypt Existing Tables

ALTER TABLE ... ENCRYPTION='Y' for all tables

Enable Log Encryption

Redo, undo, binary, relay logs

Full Encryption

3 Phase 3

Rotate Master Key

ALTER INSTANCE ROTATE INNODB MASTER KEY

Validate Backups

Test recovery with encrypted backups

Production Ready

Mydbops
Scaling Databases

Common Mistakes

MySQL

Losing Keyring File

Permanent data loss without keyring backup. All encrypted data becomes unrecoverable.

No Backup Testing

Enabling encryption without validating backup recovery process leads to critical failures.

Forgetting Binlog Encryption

Leaves replication stream and point-in-time recovery data exposed in plaintext.

No Performance Benchmarking

Deploying without testing encryption overhead results in unexpected performance degradation.

No Documented Recovery Process

Missing step-by-step recovery documentation causes prolonged downtime during critical incidents

Final Takeaways

File-Level Protection

Encryption protects against file-level attacks

Encrypt All Layers

Must encrypt tablespace, redo, undo, and binary logs

Key Management

Critical for data availability and recovery

Operational Planning

Required for successful implementation

Test Before Rollout

Validate in staging environment first

Measure Performance

Benchmark encryption overhead impact

Thank You