MongoDB Data Encryption

MongoDB offers robust encryption features to protect data while in transit, at rest, and in use–safeguarding data through its full lifecycle.

Get Started

MongoDB Security Hub

Encryption in transit

Encryption in transit secures data during transmission between clients and servers, preventing unauthorized access or tampering. In MongoDB Atlas, all network traffic to MongoDB clusters is protected by Transport Layer Security (TLS), which is enabled by default and cannot be disabled. The default version is TLS 1.2. Data transmitted to and between MongoDB cluster nodes is encrypted in transit using TLS, ensuring secure communication throughout.

MongoDB Enterprise Advanced also supports encryption in transit using TLS.

Learn more about Encryption In Transit →

Encryption at rest

Encryption at rest ensures that all stored files and data are encrypted, providing a critical layer of database-level protection. In MongoDB Atlas, customer data is automatically encrypted at rest using AES-256 to protect all volume (disk) data. This process utilizes your cloud provider's transparent disk encryption, with the provider managing the encryption keys. Additionally, you have the option to enable database-level encryption, allowing you to use your own encryption keys via AWS Key Management Service (KMS), Google Cloud KMS, or Azure Key Vault.

MongoDB Enterprise Advanced integrates at-rest encryption directly into its WiredTiger storage engine using AES-256. You can configure at rest encryption in Enterprise Advanced with a KMIP-enabled key management provider.

Encryption at Rest → MongoDB Enterprise Advanced and MongoDB Atlas

In-Use Encryption

Encryption in use secures data while it's being processed. Data is encrypted on the client side using customer-controlled keys before it's sent to, stored in, or retrieved from the database. The

throughout its entire lifecycle, whether in use, during backups, at rest, or in transit.

- **Compliance assurance:** Helps meet strict data privacy requirements such as GDPR, HIPAA, PCI DSS, and more.
- **Integrated protection for streamlined development:** In-use encryption is included with MongoDB at no extra cost, eliminating the need for third-party encryption solutions and allowing developers to work with MongoDB using built-in, familiar development patterns.

MongoDB has two features for encryption in use to meet your data protection needs: Client-Side Field-Level Encryption and Queryable Encryption.

Client-Side Field-Level Encryption

Client-Side Field-Level Encryption (CSFLE) is an in-use encryption capability that enables a client application to

encrypt sensitive data before storing it in the MongoDB database. Sensitive data is transparently encrypted, remains encrypted throughout its lifecycle, and is only decrypted on the client side.

Learn more → Client-Side Field-Level Encryption

Queryable Encryption

Queryable Encryption is a first-of-its-kind in-use encryption technology that helps organizations protect sensitive data when it is queried and in use on MongoDB. It allows applications to encrypt sensitive data on the client side, securely store it in the MongoDB database, and perform equality and range queries directly on the encrypted data. This ensures strong cryptographic protection for sensitive information without sacrificing the ability to perform expressive queries on it.

Additional benefits you can get with Queryable Encryption:

- **Groundbreaking technology:** Queryable Encryption introduces an industry-first encrypted search algorithm using NIST standards-based primitives such as AES-256,

SHA2, and HMACs. Developed by the MongoDB Cryptography Research Group and unmatched in the industry, this innovation leverages their decades of pioneering expertise in cryptography and encrypted search.

- **Expressive query capabilities on encrypted data:** Equality and range queries can be performed on encrypted data with prefix, suffix, and substring query capabilities planned.
- **Enhanced regulatory compliance:** Keep data encrypted throughout its lifecycle to ensure compliance with regulations like GDPR or HIPAA, avoiding costly fines and legal issues while boosting customer confidence.
- **Diverse use cases unlocked:** Queryable Encryption significantly reduces the risk of data exposure for organizations and improves developer productivity by providing built-in encryption capabilities for highly sensitive application workflows–such as searching employee records, processing financial transactions, or analyzing medical records–with no cryptography expertise required.
- **Improved operational efficiency:** Maintain high levels of security for your sensitive data without compromising on

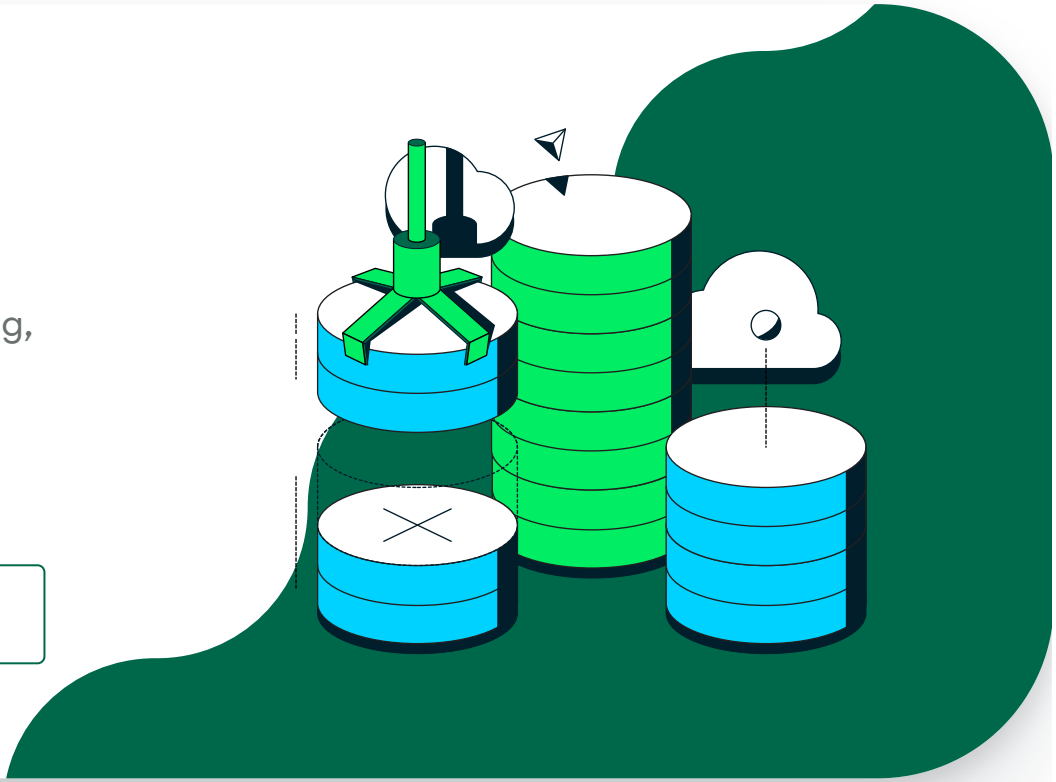
application performance or developer productivity.

Learn more → **Queryable Encryption**

MongoDB 8.0

With 36% higher throughput, easier horizontal scaling, and expanded queryable encryption, MongoDB is faster and more secure than ever.

Learn More



Resources

Queryable Encryption is generally available

Details on Queryable Encryption technology and customer benefits.

Read the blog >

Encryption at rest in Atlas using customer key management

Configure encryption at rest with your encryption keys using AWS KMS, Google Cloud KMS, Azure Key Vault.

Read the documentation >

Encryption at rest (Enterprise)

Learn more about the encryption process and how to configure encryption at rest.

Read the documentation >

Client-Side Field-Level Encryption

Learn more about how to encrypt sensitive fields from the client side, how to use MongoDB drivers and more.

Read the documentation >

Cryptography Research Group

Read about the cutting-edge research and latest innovations in cryptography and privacy.

Learn more >

Queryable Encryption Technical paper

A deeper look at Queryable Encryption, its design goals, threat models, and security guarantees.

Download the white paper >

Why Queryable Encryption matters

Learn more about why Queryable Encryption matters to developers, security teams and IT decision makers.

Read the brief >

Protect your data with MongoDB's In-Use encryption

Learn about how MongoDB's in-use encryption solutions helps customers to protect their data.

Read the datasheet >

FAQ

How do I get more information to help my organization with strong technical controls?

Contact Us

How does in-use encryption compare with in-transit and at rest encryption?

In-use Encryption is best applied selectively to those fields of your documents that you classify as containing the most sensitive data, such as PII or PHI.

Using Client-Side Field-Level Encryption data encryption alongside in-transit and at-rest encryption provides data encryption throughout its lifecycle, using complementary approaches that provide a defense-in-depth security posture to address different threat models.

- In-transit encryption protects all data traversing the network but does not encrypt data in use or at rest.
- At-rest encryption protects all stored data but does not encrypt data in use or in transit.
- With in-use encryption, your most sensitive data never leaves your application in plaintext. Fields that are encrypted on the client side cannot be decrypted by the server and remain encrypted in transit, at

Do the customer provided encryption keys used for Atlas at rest encryption need to be stored in the same cloud provider as the data is?

Can I use cloud-provider KMS for Encryption at rest with MongoDB Enterprise?

How does Queryable Encryption differ from Client-Side Field-Level Encryption?

What query types are supported with Queryable Encryption?

English

About

Support

Careers

Investor Relations

Legal

GitHub

Security Information

Trust Center

Connect with Us

Contact Us

Customer Portal

Atlas Status

Customer Support

Manage Cookies

Deployment Options

MongoDB Atlas

Enterprise Advanced

Community Edition

Data Basics

Vector Databases

NoSQL Databases

Document Databases

RAG Database

ACID Transactions

MERN Stack

MEAN Stack