K2VIEW

THE LOGICAL WAY TO MANAGE YOUR DATA

DATA SECURITY

Overview

With more and more data breaches making headlines these days, enterprises are starting to take a closer look at their own security controls to ensure they are protecting one of their most valuable assets, their data.

Today, many IT departments use a variety of encryption protocols to protect their data. However, when these protocols need to be applied to a diverse IT ecosystem with hundreds, or even thousands of systems, the process can be cumbersome and lead to an all-or-nothing approach. Meaning, if the security controls for a particular system are breached, all the data in that system will be available for the taking.

What the market needs is a customizable security solution with robust controls to improve the administration and security of the data in a way that eliminates a chance for a large-scale data breach.

K2VIEW FABRIC

K2View Fabric is a modern and highly secure distributed data management platform that secures and liberates data from your existing systems and turns it into a dynamic resource that moves at the speed of your business.

At the Heart of K2View Fabric: The Logical Unit

Most database management systems store data in silos, based on the type of data being stored (e.g. customer data, financial data, address data, device data). When data is needed, hundreds or thousands of tables may need to be queried using complex joins to deliver the information. The process can be very cumbersome, complex and time consuming.

K2View organizes and stores data differently. By storing data based on the needs of the business and not on a pre-defined structure, enterprises can improve the speed, agility and with security of their data.

Any business-related entity (a customer, a product, a service, etc.) can be represented by a Logical Unit. A Logical Unit is associated with a schema that defines all the relevant pieces of data associated with the defined business entity. The schema creation process is simple and can be done using the Auto-Discovery wizard, which enables automatic generation of the Logical Unit schema based on predefined databases constraints (primary key, foreign key), or by manually using a user-friendly drag and drop configuration console.

The result of the schema creation process is a business-oriented structure (Logical Unit) containing all the data from every table and object from every system (e.g., 100 tables from the CRM system running MySQL with 200 tables from the billing system residing on Oracle).

Managing data as logical, compressed and encrypted micro-databases delivers incredible performance, enhanced security, high availability, and customizable data synchronization.

Whether your data resides on premises, in the cloud, or in a hybrid environment, the Logical Unit concept is a bridge between scattered, hard to maintain data and highly available, secure, business-oriented data that can be delivered in real-time to any person, application, or device.

Logical Unit - Data Encryption

Fabric encrypts each Logical Unit Instance (LUI) using the AES256 algorithm in OFB mode and creates a unique encryption on each LUI. The data of each LUI is only readable using the master key. The master key is always encrypted. This additional atomic-level encryption delivers greater protection for sensitive data and eliminates the chance of a large-scale data breach.

Fabric Hashing Mechanism

The Fabric Masking mechanism uses the SHA-256 algorithm for hashing data. For example:

- An Instance Key used to encrypt LU Instances is the hashed combination of the LU name, LU Instance, and the master key.
- Fabric masking utility uses the SHA-256 algorithm when hashing the original value.

Protection Key

The AES-256 protection key is used to encrypt the master key and is stored in the Keystore, which is locked by a password.

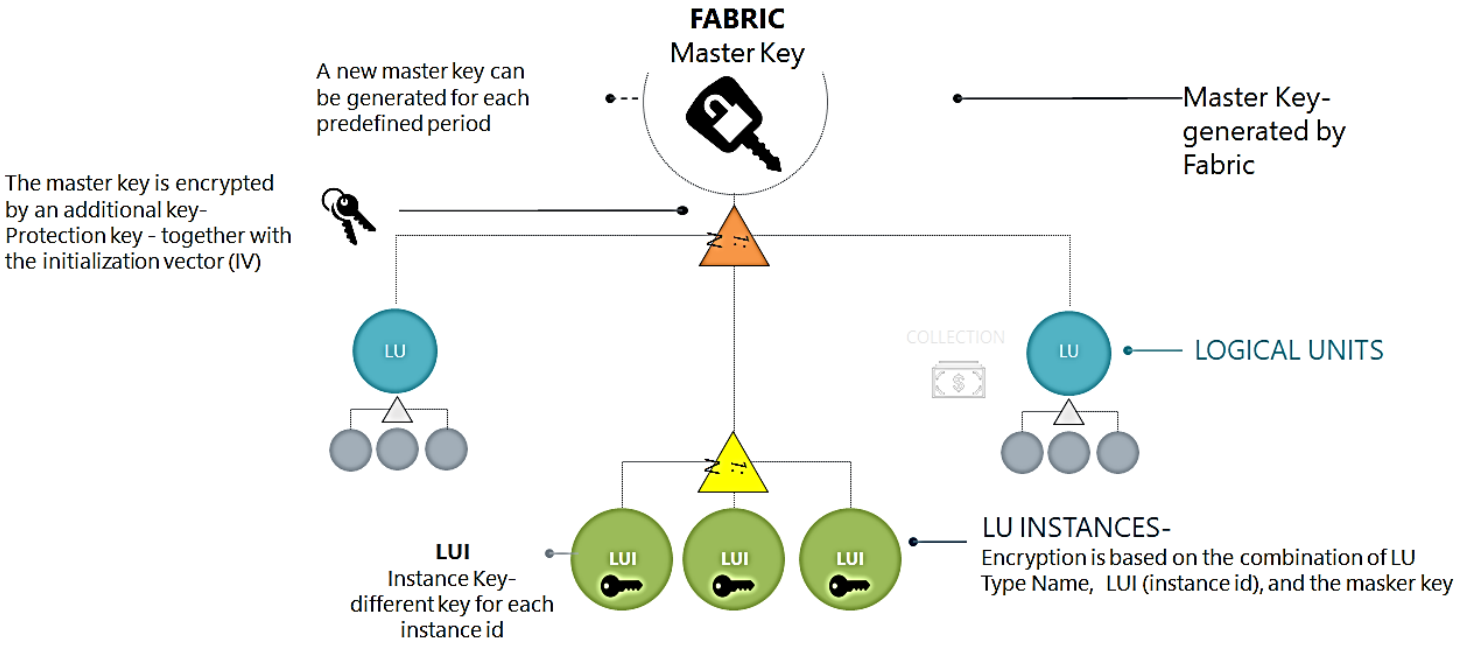
Master Key (encryption/decryption)

A unique, encrypted AES-256 key is generated by Fabric and can change from time to time. The master key is broken into bytes, where each byte is stored in a separate record in a dedicated Cassandra table. Fabric generates a key description which is a logical identifier for each master key.

Instance Key

The instance key is used to encrypt LUI. Each instance key is unique and is hashed via the SHA-256 algorithm that creates a combination of the LU name, LU Instance, and the master key.

Data Encryption Flow



WRITE

1. User A gets the LUI data from a source DB. The LUI data is encrypted by Fabric.
2. The Logical Identifier of the master key (encryption key) is attached to the LUI and identifies the master key used by Fabric to encrypt the LUI.

READ

1. User B gets the encrypted LUI from Fabric. The master key's Logical Identifier is attached to the LUI. Fabric can decrypt the master key using the Logical Identifier.
2. Fabric decrypts the data based on the decrypted master key.

LU Instance - Partial Encryption

Fabric enables users to encrypt only selected fields with sensitive data from one LUI instead of all the data from all LU instances.

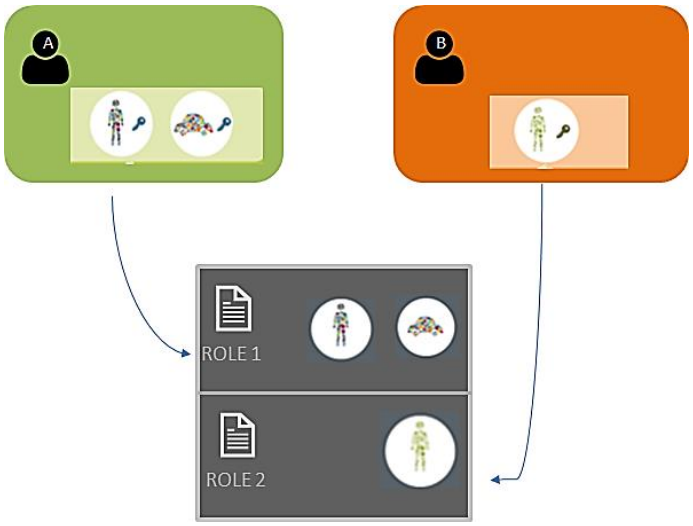
Encrypting DB Interface Details

Users can define several source environments with customized interfaces and switch between the environments when needed. Fabric encrypts passwords to secure the access to source environments. Encryption is based on the encrypted master key that encrypts the data of the LU Instances.

User Access Control

K2view Fabric allows users to control the access to data, based on user roles. Permissions can be set on the LU type level or instance level. This following example describes how different users can access different Logical Units.

- User A is attached to 'Role 1' which has access to two Logical Units.
- User B is attached to 'Role 2' which has access to a third Logical Unit.



ACCESS METHOD

Fabric offers full control over role definitions by allowing Admin users to control the access to Logical Unit types or instances. Admin users can:

- Control the Logical Unit type or instance to allow users to read or write over its structure.
- Restrict access by method, for example by using a Web Service reading method.

Secure Inter-Node Connections Using TLS/SSL Connections

Cassandra provides secure communication between a client machine and a database cluster, and between nodes within a cluster. Enabling encryption ensures that data in flight is not compromised and is transferred securely. The options for client-to-node and node-to-node encryption are managed separately and may be configured independently.

In addition to the Cassandra SSL mechanism, K2view Fabric uses its own SSL mechanism for the traffic between the Fabric nodes. In order to set SSL between the Fabric nodes, you need to configure Fabric accordingly.

FREQUENTLY ASKED QUESTIONS

How do you maintain access when modifying roles?

When a permission is added to a role, the updated permission sets impact all the users that are associated with the role. The update of permissions does not require downtime of the Fabric cluster.

Will a super user always have full access to data in K2View Fabric?

No, with K2View Fabric, you can limit the access of administrative users, so they do not have access to any Logical Units.

What is the main difference between security in K2View Fabric versus other data management solutions?

K2View Fabric is the only solution that restricts and encrypts access at the LUI level and allows users to define row-level security for the highest level of data protection, where each instance id has its own unique encryption.

CONTACT INFORMATION

www.k2view.com

info@k2view.com