AUTHOR
Martin Hentschel
Peter Povinec

SEP 18, 2015

Encryption Key Management in Snowflake

Product and Technology Cybersecurity

SHARE

SUBSCRIBE

Data encryption is one of the pillars of service offerings in the cloud. Customers demand, and at times require, that their data be fully encrypted using latest security standards. At Snowflake, all customer data in our cloud data warehouse service is encrypted by default, using latest security standards and best practices, at no additional cost. In this blog post we take a deep dive into one area of Snowflake's data warehouse security practices: encryption key management within the Snowflake security framework.

Understanding how data is encrypted in a data warehouse allows customers to better evaluate different products and to build trust between the customer and the data warehouse provider. As laid out in our security whitepaper, Snowflake uses strong AES 256-bit encryption with a hierarchical key model rooted in AWS CloudHSM. Keys are automatically rotated on a regular basis by the Snowflake service, and data can be automatically re-encrypted ("rekeyed") on a regular basis. Data encryption and key management is entirely transparent to the customer and requires no configuration or management.

To better understand how the Snowflake data warehouse encrypts customer data, we explain in detail three important aspects of Snowflake's encryption key management:

1. Hierarchical Key Model
2. Key Rotation
3. Rekeying

We will conclude by showing how AWS CloudHSM fits into this picture and why it completes Snowflake's encryption key management.

Hierarchical Key Model

A hierarchical key model is the cornerstone of Snowflake's encryption key management. A key hierarchy has several layers of keys where each layer of keys (the parent keys) encrypts the layer below (the child keys). When a key encrypts another key, security experts refer to it as "wrapping". In other words, a parent key in a key hierarchy wraps all of its child keys.

START FOR FREE

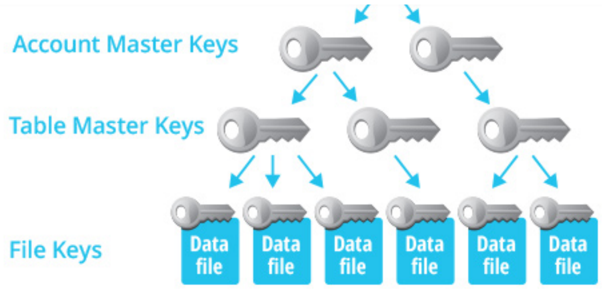

Figure 1: Snowflake's hierarchical key model.

Snowflake's hierarchical key model consists of four levels of keys: the root key, account master keys, table master keys, and file keys (Figure 1). Each account master key corresponds to one customer account in Snowflake. Each table master key corresponds to one database table in a database. That means that every account and every table is encrypted with a separate key. Similarly, every single data file is encrypted with a separate key.

A hierarchical key model provides more security for customers in a multi-tenant cloud service. With such a model, all customer accounts are isolated from each other because each account has a separate account master key. We designed this model explicitly to isolate customer accounts in our multi-tenant data warehouse. Note that this layer of isolation is used in addition to the access-control layer, which separates and isolates storage of customer data at Snowflake.

Hierarchical key models are good practice in general because each layer of keys reduces the scope of their applicability. For example, table master keys reduce the scope of their applicability to single tables; file keys further reduce the scope of applicability to single files. Thus, a hierarchical key model is essential to constrain the amount of data each key protects, and the duration of time during which it is usable (see next section).

Encryption Key Rotation

Encryption keys go through different states during their life cycle. In the active state, the key is used to encrypt data; it is available for usage by the originator. In the retired state, the key is only used to decrypt data; it is available for usage by the recipient. When a key is destroyed, it is not used for either encryption or decryption. Key rotation ensures that keys go from the active state to the retired state in this life cycle during a limited period of time.

Key rotation creates new versions of higher-level keys at regular intervals. After a specific time interval, a new version of a key is created and the previous version of this key is retired. The new version of the key is used to encrypt data. The previous version of the key is retired and only used to decrypt data. When wrapping child keys in the key hierarchy, or when inserting data into a table, only the current, active key is used to encrypt data. In other words, with key rotation, "new data gets fresh keys".

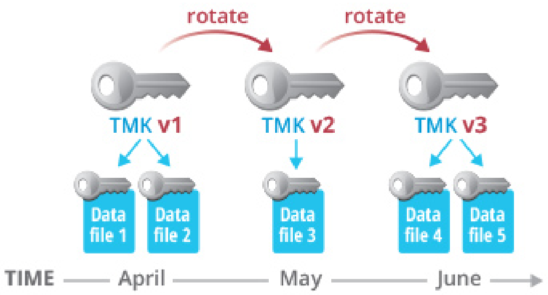

Figure 2: Key rotation of one table master key (TMK) over a time period of three months.

Figure 2 illustrates key rotation of one table master key (TMK). A table master key protects a single table in the Snowflake data warehouse. In Figure 2, the TMK with version 1 is active during April. Data that is inserted into this table in April is protected with this TMK v1. In May, TMK v1 is rotated: TMK v1 is retired and a new TMK v2 is created, a fully new random key. Version 1 is now only used to decrypt data from April. New data that gets inserted into the table is encrypted using TMK v2. In June, TMK v2 is rotated: TMK v2 is retired and a new TMK v3 is created. TMK v1 is used to decrypt data from April, TMK v2 is used to decrypt data from May, and TMK v3 is used a key is protecting. Limiting the lifetime of a key is a NIST-recommended practice to enhance security.

Rekeying

Rekeying is an additional security feature provided by Snowflake. Rekeying is the process of re-encrypting data with new keys. After a specific time interval, data that has been encrypted with an old key gets re-encrypted with a new key. This is independent and completely orthogonal to key rotation. While key rotation ensures that a key is transferred from its active state (originator usage) to the retired state (recipient usage), rekeying ensures that a key is transferred from its retired state to being destroyed. In other words:

> Key rotation = “new data gets fresh keys”
>
> Rekeying = “old data gets fresh keys”

Rekeying, therefore, completes the life cycle of keys by ensuring that keys can be destroyed.

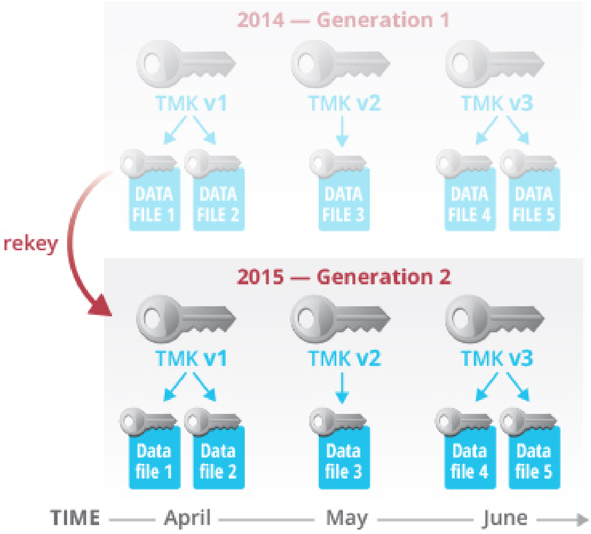

Figure 3: Rekeying one table master key (TMK) after one year.

Figure 3 illustrates rekeying in combination with key rotation. The TMK in this figure is rotated every month, as was explained in the previous section. In addition, the TMK in Figure 3 is rekeyed after one year. That is, in April 2015, TMK v1 is rekeyed. A new generation 2 of TMK v1 is created, a fully new random key. The data files protected by TMK v1, generation 1 are decrypted and encrypted with TMK v1, generation 2. Because all data files are now protected with a new TMK, the old TMK v1, generation 1 can be destroyed; it is not used anymore. In this example, the life cycle of a key is limited to a total duration of one year.

The benefit of rekeying is that it constraints the total duration during which a key is used for recipient usage. This, again, increases security as recommended by NIST. Furthermore, when rekeying data, it is possible to increase encryption key sizes and utilize better encryption algorithms that may be invented and standardized in the future. Rekeying therefore ensures that all customer data, new and old, is encrypted with latest security technology.

Snowflake’s unique capability allows rekeying data files online, in the background, without any impact to currently running customer workloads. Data that is being rekeyed is always available to the customer. No downtime of the service is necessary to rekey data and no performance impact is observed on the customer workload. This is a unique capability of the Snowflake service and a direct result of Snowflake’s architecture of separating storage and compute resources.

Amazon CloudHSM

To complete Snowflake’s encryption key management, we use AWS CloudHSM (online hardware security modules) as a tamper-proof, highly secure way to generate, store, and use the root keys of the key hierarchy (Figure 4). Using CloudHSM provides the following security benefits:

> Wrapped lower-level keys in the key hierarchy cannot be unwrapped without authorized access to the HSM devices.
>
> Finally, in addition to generating new encryption keys when creating new accounts and tables, CloudHSM generates secure, random encryption keys during key rotation and rekeying.

We use CloudHSM's high-availability configuration with an additional offline backup device to reduce the possibility of service outages and **to be safe from losing the hierarchy's most important keys**.

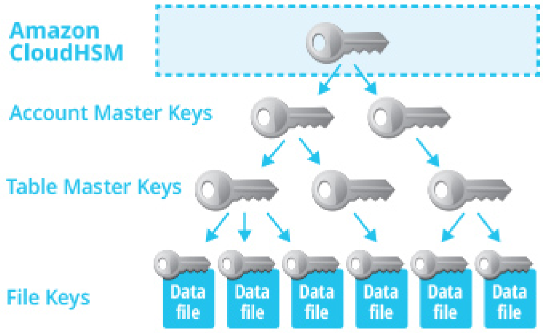

Figure 4: Key hierarchy rooted in Amazon CloudHSM.

Data Security – As a Service

These components of encryption key management and usage are only one part of a comprehensive security strategy. If you had to handle all of the different components and processes yourself, it would be a significant burden on your already-stretched team.

At Snowflake we've taken the approach of implementing and maintaining a full array of security features and controls as a built-in part of our data warehousing service. Our customers can be assured that their data is secured using best-in-class approaches for data encryption, key management, authentication, access control, and more.

As we've detailed above, Snowflake provides best-in-class key management as part of our service– a hierarchical key model rooted in AWS CloudHSM, automatic key rotation, and even rekeying. All of that is entirely transparent to the customer and requires no configuration, management, or downtime. That's data security as a service.

SHARE

Data Encryption with Customer-Managed Keys

For customers with the highest security requirements, Snowflake offers customer-managed keys.

Find Out More

ETL Testing for the Data Warehouse

Testing is necessary to ensure that data moving is accurate at each point. Learn how to minimize impact of ETL testing and ETL alternatives in the Data Cloud.

Find Out More

Data Encryption with Customer-Managed Keys for Azure

In 2017, Snowflake announced support for customer-managed keys using AWS Key Management Service (KMS). These keys, created...

Find Out More

How Does
Work?

Learn about d
is evolving to
modern cloud-

Discover

Snowflake Inc.

PLATFORM

Cloud Data Platform
Architecture
Pricing
Marketplace
Security & Trust

SOLUTIONS

Snowflake for Financial Services
Snowflake for Advertising, Media, & Entertainment
Snowflake for Retail & CPG
Healthcare & Life Sciences Data Cloud
Snowflake for Marketing Analytics

RESOURCES

Resource Library
Webinars
Documentation
Community
Procurement
Legal

EXPLORE

News
Blog
Trending
Guides
Developers

ABOUT

About Snowflake
Investor Relations
Leadership & Board
Snowflake Ventures
Careers
Contact