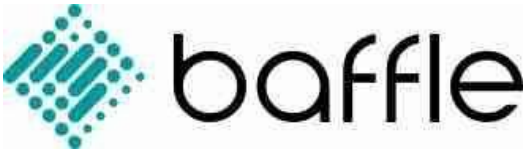

A Technical Overview of Baffle Hold Your Own Key (HYOK) and Record Level Encryption (RLE)

Introduction

Baffle Data Protection Services (DPS) provide a range of data encryption, tokenization and de-identification methods to protect data in data stores and cloud storage environments. Common methods that Baffle employs include column or field level encryption, tokenization, format preserving encryption (FPE), dynamic data masking, record or row level encryption (RLE), and privacy preserving analytics.

It is important to note that Baffle DPS is a software solution that allows organizations to deploy their own data protection service layer on-premise or in the cloud with their own keys – in other words, Baffle is NOT a SaaS solution.

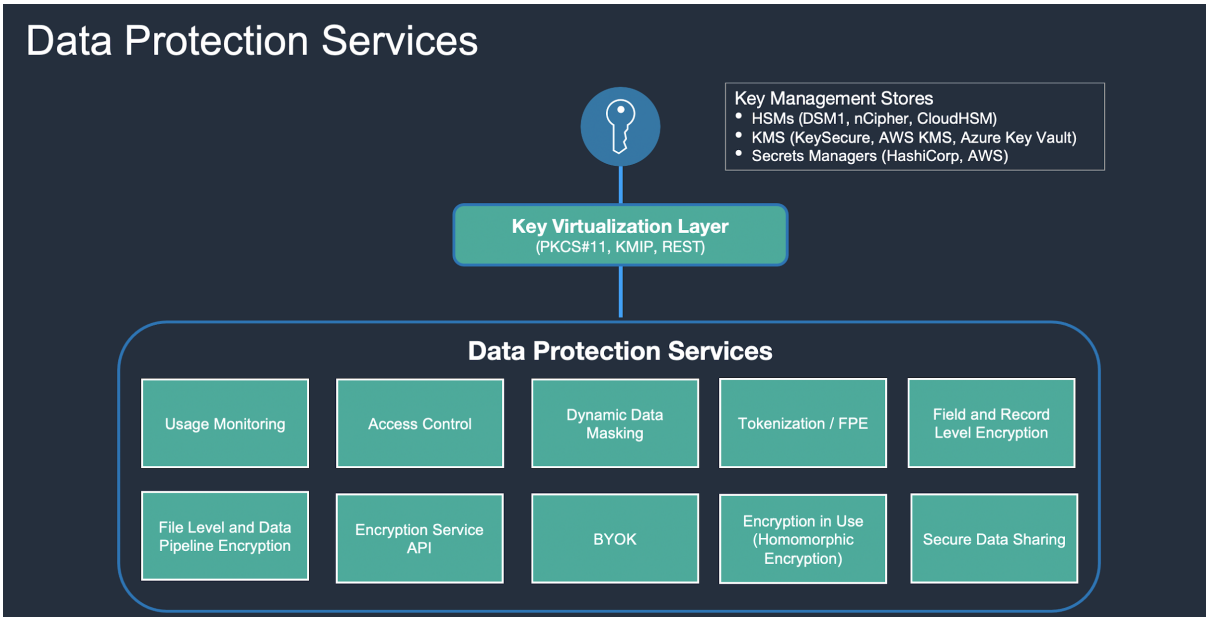

Figure 1 - An overview of different capabilities available with Baffle DPS

Baffle also integrates with key management stores via a key virtualization layer and has extended this capability to support multiple master encryption keys (e.g. MK, CMK, KEK). This latter method of key support enables a Bring Your Own Key (BYOK) or Hold Your Own Key (HYOK) encryption key model for multiple parties where data may be co-mingled in a data store.

This paper provides a technical overview of Baffle's HYOK implementation and how it can be applied to provide RLE in multitenant or shared data stores.

Encryption Key Management and Two-Tier Hierarchy

A significant challenge in facilitating HYOK at-scale involves incorporating client key management functionality into the application stack. Baffle's method of integrating with key management stores and hardware security modules (HSMs) employs an envelope encryption or two-tier key management hierarchy.

Envelope encryption utilizes a master key (MK) or a key encrypting key (KEK) and uses it to encrypt data encryption keys (DEKs). The DEKs are mapped to key identifiers which are then used to encrypt a specific object, value, column or row. Using this model, master keys can be rotated and used to re-encrypt DEKs, which will address security and compliance concerns for most environments.

Baffle's solution uses industry standard protocols such as Key Management Internet Protocol (KMIP), PKCS #11 or REST APIs to communicate with various key management solutions. Baffle supports integration with numerous key stores including the following (partial list):

- AWS KMS
- AWS CloudHSM
- Azure Key Vault
- HashiCorp Vault and Consul
- IBM Key Protect
- Thales KeySecure

Enabling HYOK for Multiple Parties

Implementing column level or application level encryption and integrating with key management solutions can present significant challenges for application development teams. Enabling this type of encryption at-scale in a dynamic infrastructure introduces additional hurdles, and when one looks to extend the model to SaaS or multi-tenant environments with multiple data owners, it creates further complexity.

Baffle DPS supports an HYOK environment by extending its key virtualization layer to support multiple disparate master keys. In this type of deployment, a master key is mapped to an entityID (or data owner) and used to encrypt respective DEKs which are then used to encrypt the records owned by that entityID.

Below is an example of an integration with AWS Key Management Service (KMS).

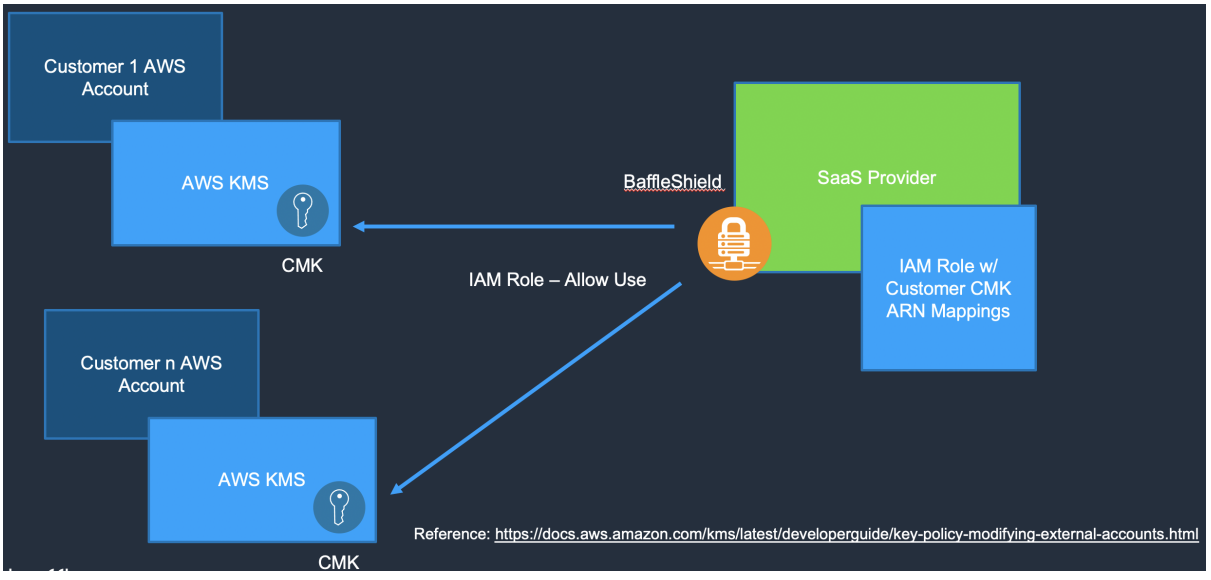

Figure 2 - An example diagram of multi-party HYOK for a SaaS environment

This model allows for customers of a SaaS provider to use a CMK from their own KMS instance and the SaaS provider sources the key and encrypts the respective DEKs using said CMK. Data in the SaaS environment is seamlessly encrypted and decrypted at the row level using a Baffle Shield proxy. If the customer chooses to disable or delete the CMK, the ability to decrypt data in the SaaS environment is disabled and access to the data is effectively revoked.

Encrypting Data at the Record Level

In order to encrypt data using HYOK, Baffle DPS is able to source keys from multiple key stores by integrating with key management stores as described above. The function of actually encrypting the data at the record or row level is most often performed by the Baffle Shield. The Baffle Shield is a stateless wireline proxy for databases and performs encryption and decryption using the sourced keys. The section provides a walk-through of the Baffle Shield architecture and functionality.

[NOTE: Baffle also provides a REST API and SDK which can be used in addition to its core Baffle Shield]

The Baffle Shield operates in a stateless manner and never persists any data or encryption keys. It does leverage the key virtualization layer described above to source keys and apply them to rows based on key and entity mappings. In order to operate, the Baffle Shield maintains a privacy schema which contains knowledge of the data schema and fields to be encrypted. It also holds an entity schema which maintains a mapping of the entityIDs and associated master keys.

When Baffle Shields are deployed, the application connection string is changed to point to the Baffle Shield endpoint. Queries for data issued by the application are seamlessly parsed by the Baffle Shield and encrypt and decrypt operations are performed based on the privacy and entity schemas.

	psk	name
1	101	¾¾¾¼ ·³ôPÔ¦fR&Ä9Q Éá[TUÚ < Ï... [38]
2	102	¾¾¾¼ Ú¬på-q ÕO Qòs7f¦b:½Ûk áh... [37]
3	103	¾¾¾¼ b_÷BHLý¾ñ« é Xû¢ðJ¤þ©\ í... [41]
4	104	¾¾¾¼ w:¤ dL© kò äÖ¡ Ð¶Á%(iÂ... [39]
5	105	¾¾¾¼ a¯{¢ê }ÿb3pü ÀÌ$lK« d»ì... [43]
6	106	¾¾¾¼ Â`*1ÄH¢± Üô U ×±UVç¿ o ... [44]
7	107	¾¾¾¼ 6x §D]ù0 V´0ëÕ¦æz)!¡ L+... [39]
8	108	¾¾¾¼ Ì¨ Th ùoåÒ ,© v±pØ... [38]
9	109	¾¾¾¼ k Èl»>Qg × þ«l A)}Ü]3 ... [38]
10	110	¾¾¾¼ evÿ6w] 5 ¿§Á}1 ¸EÅ« àn... [43]
11	111	¾¾¾¼ É~|p[+ë .LÕ*7W Å@ z oø... [38]
12	112	¾¾¾¼ F. ´ü ± Åá t$ Ú(Ô]... [41]
13	113	¾¾¾¼]zÕ [l vofð)Ûú×SýåE¨ A... [37]
14	114	¾¾¾¼ ß¤c+)R ×ÎôF FÔÀcHG õ§ $... [37]
15	115	¾¾¾¼ UX CÖï¾Ý#è r·ÈSäx ë¼ ... [40]
16	116	¾¾¾¼ ¯Ïþq % WmE¦ » wÈÙOS... [41]
17	117	¾¾¾¼ î)@ ² O¹>Ày z jsÉ0 _... [43]
18	118	¾¾¾¼ ÝÂl ¥Æ+wÜF ßì]õ(q 2D»... [41]
19	119	¾¾¾¼ çkR ´¥ àg Ö y; ½ÿÈ ñD&... [37]
20	120	¾¾¾¼ Evôþå6£Õ 7 S27 ,|Ä Z@|x... [39]
21	121	¾¾¾¼ ´¨ <Ì8ØñÙ¼Ú6²ß`ã ç§£º®#... [41]
22	122	¾¾¾¼ õ" HâYÊhHLBû61Ù"É# | ø ... [41]
23	123	¾¾¾¼ +î\éÔ¦ º Tðí m{C! #¢Uïm... [41]

Figure 3 - Data encrypted in the 'name' column at the row level

The above picture shows a column ‘name’ encrypted at the row level. All the of the data is indecipherable, as this data is viewed via direct access to the database.

Conversely, by passing session context to the Baffle Shield, the same query is sent, but leverages the privacy and entity schema to source the appropriate master key and DEK to decrypt the data for an authorized party. In the example below, select rows are available for decryption by the respective data owner. The rest of the data is dynamically masked, however, the result set can also be suppressed or a query client error can be returned instead if no key is available.

	psk	name
1	101	Vallie Allen
2	102	***
3	103	***
4	104	Myrle Nicosia
5	105	***
6	106	***
7	107	Lashay Worman
8	108	***
9	109	***
10	110	Tanner Mcgranahan
11	111	***
12	112	***
13	113	Adina Riles
14	114	***
15	115	***
16	116	Clay Valderrama
17	117	***
18	118	***
19	119	Randee Hoos
20	120	***
21	121	***
22	122	Tyesha Frenette
23	123	***
24	124	***

Figure 4 - Example of record level decryption of data based on authorized access to keys

This implementation allows organizations to achieve record level segmentation of data at-scale using encryption with customer owned keys to secure the data. All of this can be implemented with no code changes or “low code” changes to the application environment. Below is a high level architectural model of a Baffle DPS deployment.

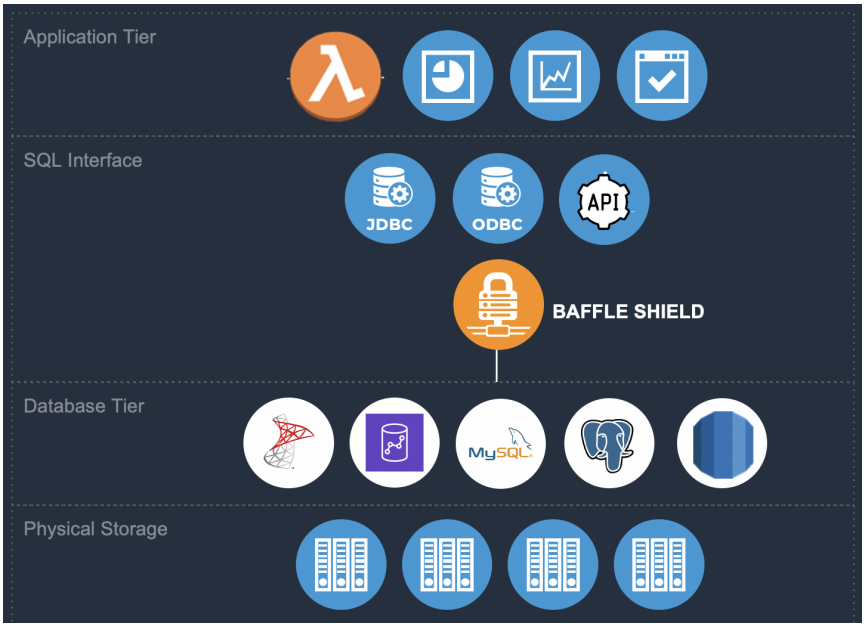

Figure 5 - High level architecture of Baffle Data Protection Services with a Baffle Shield

Deployment and Performance Considerations

Using Baffle DPS in virtually any application environment may raise questions around scalability, high availability and performance. Baffle DPS has been deployed globally inline in multi-billion record environments. The client traffic routed through the Baffle solution includes several Fortune 100 firms and thirty of the world's largest banks. The solution also supports the IOT division at a Fortune 25 manufacturer where it is inline with billions of records.

Inevitably, performance remains a primary concern for potential adopters of the solution. The main factors that affect performance are result set size and network latency, so performance will vary based on the application workload.

However, Baffle has been measured at one to two milliseconds of overhead on encrypted data payloads in production environments. The method to achieve such performance lies in the fact that the Baffle Shield has been heavily optimized. Further, clear text traffic passes through the proxy at wire speed and the privacy and entity schemas referred to above help maintain how queries are handled and parsed.

In terms of availability, the software is stateless and can effectively run anywhere. Most scaled customers deploy the solution in Kubernetes pods that have auto-scaling policies and operate in a load balanced environment. Using these methods of container-based and scaled deployments, the solution runs in 24x7 environments today. Scaling for applications is linear across the Baffle Shields and hardware or instance sizing and can accounted for predictably.

Summary

For organizations looking to address data privacy and security concerns over data in a shared data environment, Baffle DPS offers an HYOK and row level encryption capability that can simplify implementation and management of customer-owned keys and crypto operations. Many companies do not want to spend the resources or acquire the domain knowledge to build such solutions in-house. And as environments grow and scale, relying on a common architectural service layer allows new applications and enhancements to leverage the same common service instead of embedding it time and again in each application component.

Baffle DPS eliminates the need to learn about key management integrations, key generation, and seamlessly enables encryption, while giving organizations and their clients peace of mind about how their data is being secured. Learn more at https://baffle.io

Additional Resources

- Download the white paper on **Simplified Application Level Encryption**
- See record level encryption in action in our **"Ensuring Data Privacy in SaaS"** webinar
- **Encryption methods** — ALE, FLE, RLE, TDE?? What the heck's the difference?
- Schedule time with a security specialist to **discuss your requirements and how we can help**