Fine Grain Access Control for Big Data: ORC Column Encryption

Owen O'Malley
owen@cloudera.com
@owen_omalley

March 2019

HORTONWORKS®
POWERING THE FUTURE OF DATA™

Who Am I?

- Worked on Hadoop since Jan 2006
    - Worked at Yahoo on Web Search team
    - First committer added to Hadoop project
- MapReduce, Security, Hive, and ORC
- Worked on many different file formats
    - Sequence File, Avro, RC File and ORC File
- Spun Hortonworks out of Yahoo in 2011

© Hortonworks Inc. 2011 – 2019. All Rights Reserved
HORTONWORKS®

What is the Problem?

© Hortonworks Inc. 2011 – 2019. All Rights Reserved

Controlling Sensitive Data

- Some data is very sensitive
  - Personally Identifiable Information
  - Credit card
  - Medical information
- Companies run on data
  - Need controls on data
  - GDPR is a **BIG** deal

© Hortonworks Inc. 2011 – 2019. All Rights Reserved
HORTONWORKS®

What is the Problem?

- Related data, different security requirements
  - Authorization – who can see it
  - Audit – track who read it
  - Encrypt on disk – regulatory
- File-level (or blob) granularity isn't enough
  - File systems don't understand columns

© Hortonworks Inc. 2011 – 2019. All Rights Reserved
HORTONWORKS®

Requirements

- Readers should transparently decrypt data
  - If and only if the user has access to the key
  - The data must be decrypted locally
- Columns are only decrypted as necessary
- Master keys must be managed securely
  - Support for Key Management Server & hardware
  - Support for key rolling

© Hortonworks Inc. 2011 – 2019. All Rights Reserved

Partial Solutions

© Hortonworks Inc. 2011 – 2019. All Rights Reserved

Partial Solution – HDFS Encryption

- Transparent HDFS Encryption
- Encryption zones
  - HDFS directory trees
  - Unique master key for each zone
  - Client decrypts data
- Key Management via KeyProvider API

© Hortonworks Inc. 2011 – 2019. All Rights Reserved

HDFS Encryption Limitations

- Very coarse protection
  - Only entire directory subtrees
  - No ability to protect columns
  - A lot of users need access to keys
- Moves between zones is painful
  - When writing with Hive, data is moved multiple times per a query

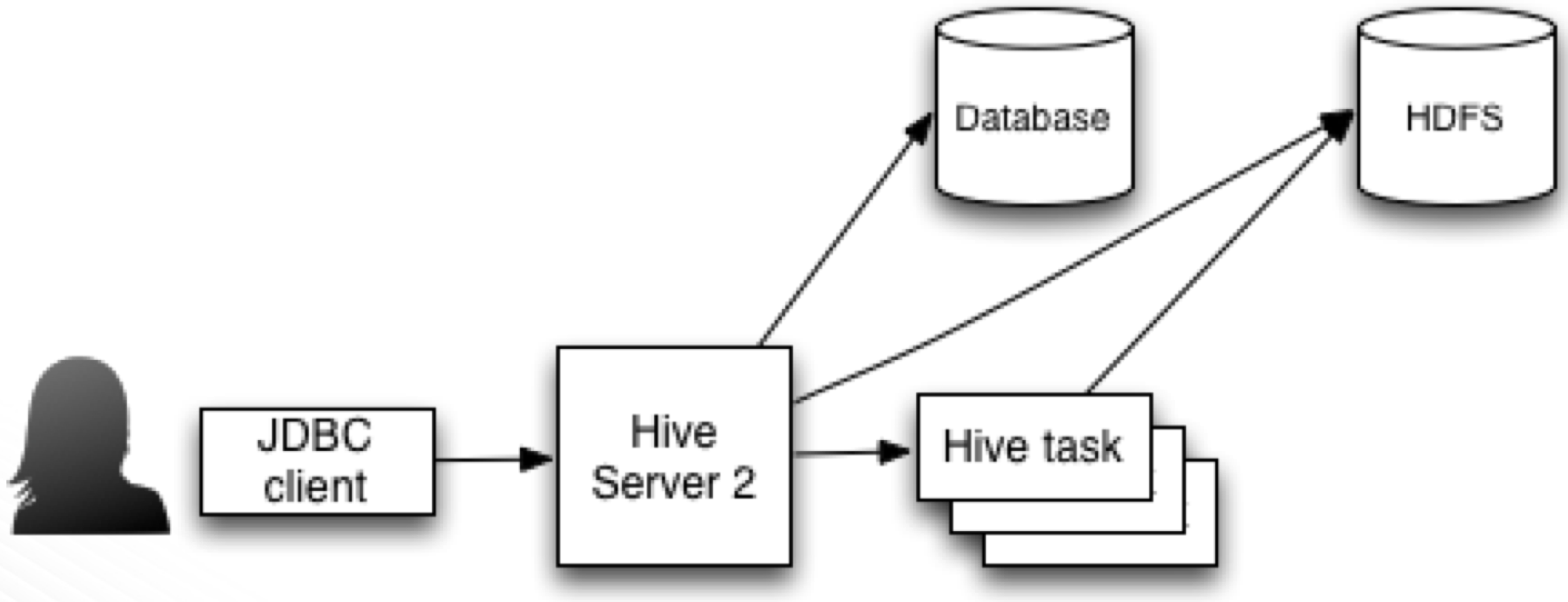
© Hortonworks Inc. 2011 – 2019. All Rights Reserved

Partial Solution – Hive Server 2

- All queries sent to Hive Server 2
  - Only ‘hive’ user has access to data in HDFS
  - Supports LLAP
- Integrates with Apache Ranger
  - Control access to rows & columns
  - Dynamically mask data

© Hortonworks Inc. 2011 – 2019. All Rights Reserved

Hive Architecture with Hive Server 2



© Hortonworks Inc. 2011 – 2019. All Rights Reserved

HORTONWORKS

Hive Server 2 Limitations

- Limits access to Hive SQL
  - Breaks Hadoop's multi-paradigm data access
  - Many customers use both Hive & Spark
- JDBC is not distributed
  - Throughput is limited to 1 machine
  - New Spark to LLAP connector addresses this

© Hortonworks Inc. 2011 – 2019. All Rights Reserved

Partial Solution – Separate tables

- Split private information out of tables
  - Separate directories in HDFS
  - HDFS and/or HS2 authorization
  - Enables HDFS encryption
- Limitations
  - Need to join with other tables
  - Higher operational overhead

© Hortonworks Inc. 2011 – 2019. All Rights Reserved
HORTONWORKS®

Partial Solution – Encryption UDF

- Hive has user defined functions
  - aes_encrypt and aes_decrypt
- Limitations
  - Key management is problematic
  - Encryption is not seeded
  - Size of value leaks information

© Hortonworks Inc. 2011 – 2019. All Rights Reserved

Solution

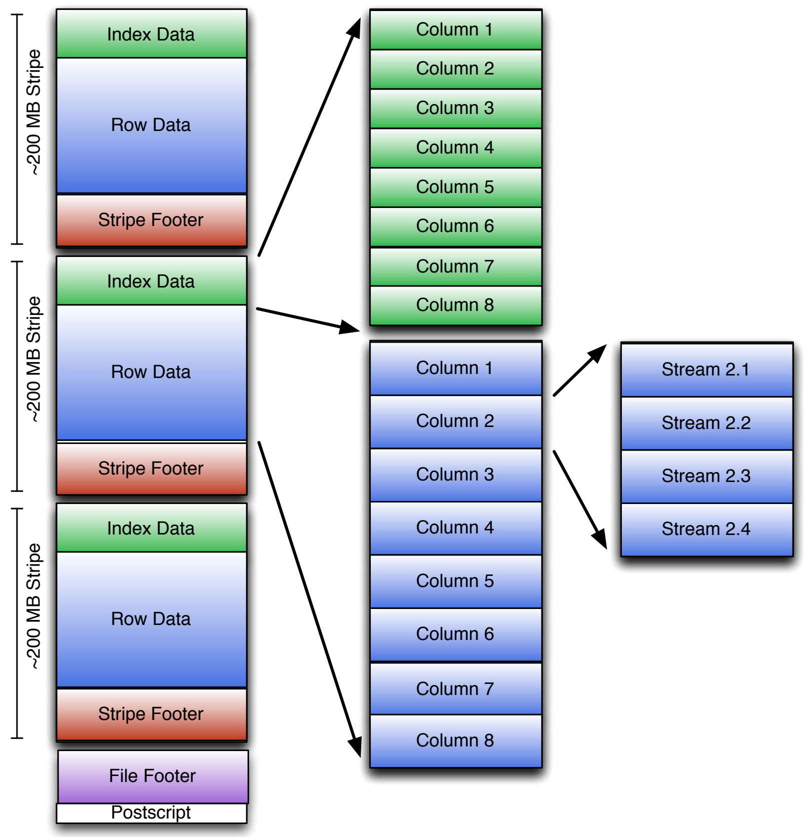
© Hortonworks Inc. 2011 – 2019. All Rights Reserved

Columnar Encryption

- Columnar file formats (eg. ORC)
  - Write data in columns
    - Column projection
    - Better compression
- Encryption works really well
  - Only encrypt bytes for column
  - Can store multiple variants of data

© Hortonworks Inc. 2011 – 2019. All Rights Reserved
HORTONWORKS®

ORC File Format



© Hortonworks Inc. 2011 – 2019. All Rights Reserved

HORTONWORKS®

User Experience

- Set table properties for encryption
  - orc.encrypt.pii = ”ssn,email”
  - orc.encrypt.credit = “card_info”
- Define where to get the encryption keys
  - Configuration defines the key provider via URI
  - Can use the Hadoop or Ranger KMS
  - Compatible with public cloud KMS

© Hortonworks Inc. 2011 – 2019. All Rights Reserved

Key Management

- Uses Hadoop or Ranger KMS
- Create a master key for each use
  - “pii”, “pci”, or “hippa”
- Each column in each file uses unique local key
- Policies limit access to master keys
  - User never gets master keys

© Hortonworks Inc. 2011 – 2019. All Rights Reserved

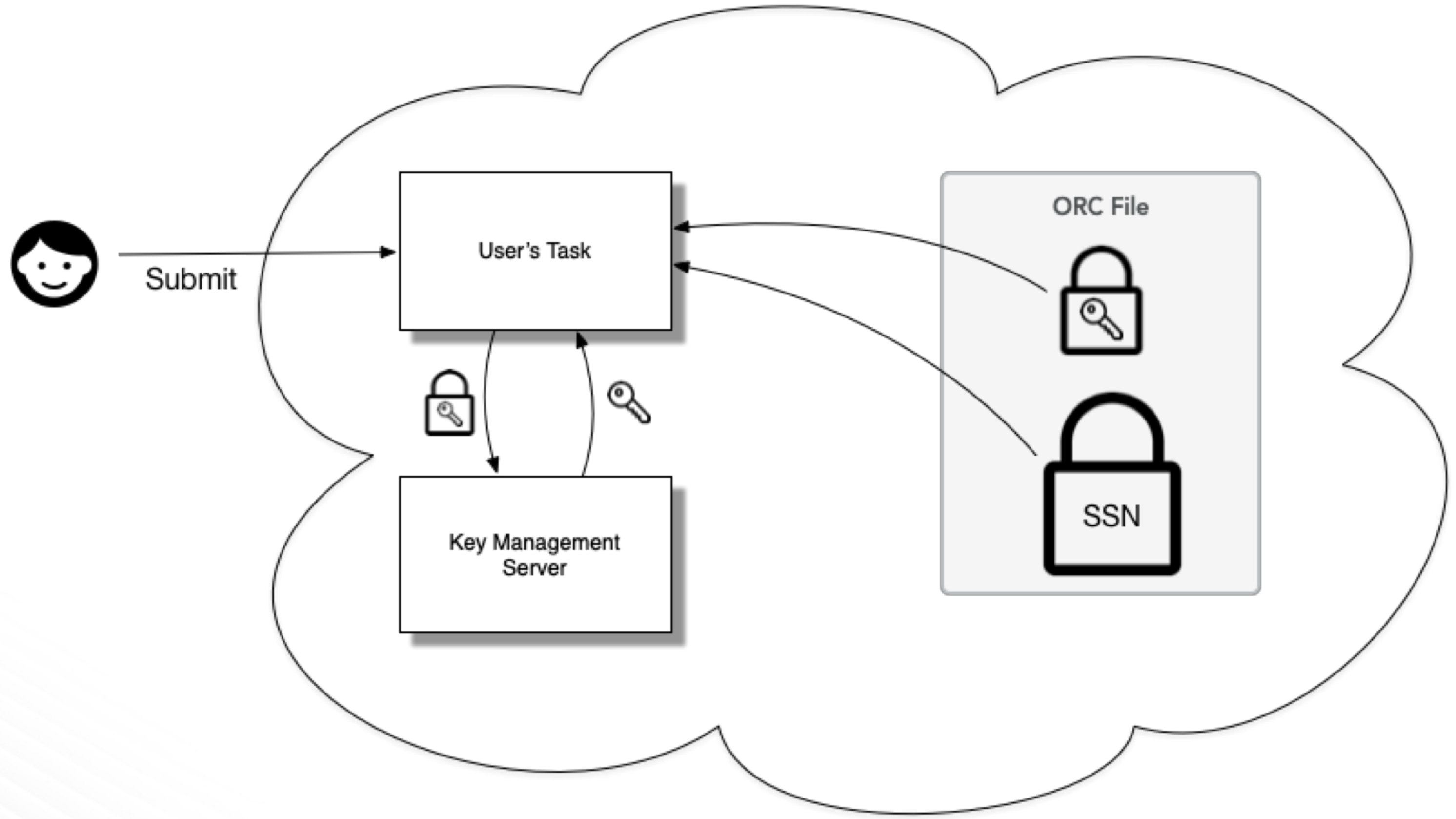
HORTONWORKS®

KeyProvider API

- Provides limited access to encryption keys
- Encrypts or decrypts local keys
- Key versions and key rolling
- Allows 3rd party plugins
    - Supports Hadoop or Ranger KMS

© Hortonworks Inc. 2011 – 2019. All Rights Reserved

Encryption Data Flow



© Hortonworks Inc. 2011 – 2019. All Rights Reserved

Encryption Flow

- Local key
  - Random for each encrypted column in file
  - Encrypted w/ master key by KMS
  - Encrypted local key is stored in file metadata
- IV is generated to be unique
  - Column, kind, stripe, & counter

© Hortonworks Inc. 2011 – 2019. All Rights Reserved
HORTONWORKS

Data Masking

- What happens without key access?
- Define static masks
  - Nullify – all values become null
  - Redact – mask values ‘Xxxxx Xxxxx!’
    - Can define ranges to unmask
  - SHA256 – replace with SHA256
  - Custom - user defined

© Hortonworks Inc. 2011 – 2019. All Rights Reserved
HORTONWORKS

Data Masking

- Anonymization is hard!
  - AOL search logs
  - Netflix prize datasets
  - NYC taxi dataset
- Always evaluate security tradeoffs
- Tokenization is a useful technique
  - Assign arbitrary replacements

© Hortonworks Inc. 2011 – 2019. All Rights Reserved

Key Disposal

- Often need to keep data for 90 days
  - Currently the data is written twice
  - With column encryption:
    - Roll keys daily
    - Delete master key after 90 days

© Hortonworks Inc. 2011 – 2019. All Rights Reserved

ORC Encryption Design

- Write both variants of streams
  - Masked unencrypted
  - Unmasked encrypted
- Encrypt both data and statistics
- Maintain compatibility for old readers
  - Read unencrypted variant
- Preserve ability to seek in file

© Hortonworks Inc. 2011 – 2019. All Rights Reserved
HORTONWORKS®

ORC Write Pipeline

- Streams go through pipeline
  - Run length encoding
  - Compression (zlib, snappy, or lzo)
  - Encryption
- Encryption is AES/CTR
  - Allows seek
  - No padding

© Hortonworks Inc. 2011 – 2019. All Rights Reserved

Conclusions

© Hortonworks Inc. 2011 – 2019. All Rights Reserved

Conclusions

- ORC column encryptions provides
  - Transparent encryption
  - Multi-paradigm column security
  - Audit logging (via KMS logging)
  - Static masking
- Supports file merging
  - Different stripes with different local key

© Hortonworks Inc. 2011 – 2019. All Rights Reserved

HORTONWORKS®

Integration with Other Tools

- Apache Ranger
  - Provides security from a single control panel
  - Provides Attribute Based Access Control (ABAC)
  - Manages encryption settings based on policies
  - Controls access to decryption keys
- Apache Atlas
  - Metadata driven governance for enterprises
  - Provides ability to tag tables or columns

© Hortonworks Inc. 2011 – 2019. All Rights Reserved

HORTONWORKS®

Integration with Other Tools

- Hive & Spark
  - No change other than defining table properties
- Apache Hive's LLAP
  - Cache and fast processing of SQL queries
  - Column encryption changes internal interfaces
  - Cache both encrypted and unencrypted variants
  - Ensure audit log reflects end-user and what they accessed

© Hortonworks Inc. 2011 – 2019. All Rights Reserved

Limitations

- Need encryption policy for write
  - Current Atlas & Ranger tags lag data
  - Auto-discovery requires pre-access
- Changes to masking policy
  - Need to re-write files
- Need additional data masks
  - Credit card, addresses, etc.
- Decrypted local keys could be saved

© Hortonworks Inc. 2011 – 2019. All Rights Reserved

Thank you!

Twitter: @owen_omalley
Email: owen@cloudera.com

© Hortonworks Inc. 2011 – 2019. All Rights Reserved