Efficient Analytics on Encrypted Data

Gidon Gershinsky
IBM Research - Haifa
gidon@il.ibm.com

ABSTRACT

Enterprises and non-profit organizations work with sensitive commercial or personal information, stored in an encrypted form due to business confidentiality requirements, GDPR regulations [4] and other reasons. Unfortunately, a straightforward encryption doesn't work well for modern columnar data formats, such as Apache Parquet [2], that are leveraged by analytic frameworks for acceleration of data ingest and processing.

Parquet is a popular file format, widely used in cloud and on-premises processing of data by Apache Spark [3], Impala [1] and other systems. Besides column-oriented information storage, Parquet enables efficient data encoding, compression and fast access to field values by use of multi-level internal indexing and statistics. The latter capability is critical for a so-called predicate push-down, where an analytic framework fetches and processes only a subset of the full data set, after analyzing Parquet metadata that narrows down the files and data pages relevant for a given query (predicate). Combined with column filtering, this allows to accelerate analytic workloads by order(s) or magnitude. However, if Parquet files are bulk-encrypted in storage, their internal modules can not be extracted and parsed. All files in a requested folder must be fully delivered from storage to the analytic framework location, decrypted and authenticated there, and then processed. Another alternative is to decrypt the files at the storage upon a read request - however, this makes the encryption keys and the data visible to the storage system and administration. Also, this still requires full decryption of every file in a folder, before the parsing becomes possible. A third option is to use an encryption client in storage SDKs, available in some clouds. But these clients don't support authentication encryption for range reads, required for predicate push-down, and make the solution tied to a specific cloud storage, inapplicable in other clouds or on-premises data centers.

We are working on a Parquet modular encryption mechanism [5] that supports *authenticated data encryption and efficient filtering in any storage, without revealing the encryption key or the data to the storage system.* The mechanism preserves the Parquet encoding, compression, columnar projection and indexing capabilities. It uses the internal modular structure of the format for a separate encryption of all data and metadata components, while updating the module references as required by authenticated encryption algorithms that don't preserve the data length. Authentication support allows a reader to make sure a file has not been tampered with or replaced with an old version. We work with the Apache Parquet community to contribute this mechanism to the open source project. Initially, the mechanism will enable a single encryption key for each file, with a choice of columns to be encrypted and columns to be left as plaintext if they don't contain any sensitive data. Later, this approach will be extended to a key-per-column and possibly key-per-rowgroup encryption. In parallel, we integrate Apache Spark with the Parquet modular encryption mechanism - to *enable Spark to work directly with encrypted data.* The integration allows for an efficient Spark SQL analytics not only on cleartext Parquet files, but on encrypted data as well.

CCS CONCEPTS

• **Security and privacy → Management and querying of encrypted data**;

KEYWORDS

data encryption, big data analytics

REFERENCES

[1] Apache Software Foundation. 2018. Apache Impala. https://impala.apache.org/. (2018).
[2] Apache Software Foundation. 2018. Apache Parquet. https://parquet.apache.org/. (2018).
[3] Apache Spark. 2018. Spark SQL and DataFrames. https://spark.apache.org/sql/. (2018).
[4] EU. 2018. General Data Protection Regulation. https://gdpr-info.eu/. (2018).
[5] Gidon Gershinsky. 2018. Parquet Modular Encryption Jira. https://issues.apache.org/jira/browse/PARQUET-1178. (2018).

SYSTOR '18, June 4–7, 2018, HAIFA, Israel

ACM ISBN 978-1-4503-5849-1/18/06…$15.00
https://doi.org/10.1145/3211890.3211907