multiple entities are involved. For example, two companies may have a need to exchange sets of documents stored in internal storage. Each company may use the same numbering system for identification, which may result in multiple documents having conflicting identifiers. As a result, either the documents must be stored in separate tables and/or databases, or identifiers must be modified for one or both of the sets. In the former instance, it may be difficult to make adjustments to both tables/databases at once and may be inconvenient for the purposes of sorting and storage. In the latter instance, the two parties must both agree on the alternative identification with one or both parties modifying their previous identification system accordingly, which can be highly inconvenient and potentially detrimental due to the prevailing use of the identifiers in the rest of their business. In addition, any future changes to identifiers, such as due to changing in business practices, or in an instance where a deal is struck with a third entity with whom there are conflicting identifiers, must be propagated to the other entity. As entities communicate and deal with more and more entities, the conflicts may compound where changes to identifiers may occur at an unreasonable rate and negatively affect each entity's ability to do business.

Thus, there is a need for a technical solution to enable a universal distributed data storage where every data set, regardless of entity, type of data, method of storage, etc. can be combined without any collisions in identification, and where the integrity is not affected by future additions of data to the data set.

SUMMARY

The present disclosure provides a description of systems and methods for generation of a universal distributed data storage. The use of collision-resistant hashing to develop identification values for disparate data sets may enable the combination of data sets without collisions and may preserve integrity of the data sets and their identification following any future combination of additional data sets into the universal data storage.

A method for generating a universal distributed data storage includes: receiving, by a processing server, a data signal superimposed with a plurality of data values from two or more data sets to be combined into a single data storage without collisions; generating, by the processing server, a key for each of the plurality of data values, wherein the respective key is generated via application of a collision-resistant hashing algorithm to an associated data value; executing, by the processing server, a first query on a database

of the processing server to store each of the plurality of data values and an associated generated key in the database of the processing server as a plurality of key-value pairs; receiving, by the processing server, a data signal superimposed with a plurality of additional key-value pairs, executing, by the processing server, a second query on the database of the processing server to store each of the plurality of additional key-value pairs in the database of the processing server, wherein each key included in the plurality of additional key-value pairs is generated via application of the collision-resistant hashing algorithm to the associated value; receiving, by the processing server, from an external system via a communication network, a data signal superimposed with a rehash request that indicates a hashing algorithm different from the collision-resistant hashing algorithm, and without requiring exchanging the data values or associated keys, generating, by the processing server, a new key for each of the plurality of data values and each value included in the plurality of additional key-value pairs by rehashing the data values using the hashing algorithm received from the external system.

A system for generating a universal distributed data storage includes: a database of a processing server; a processing server configured to receive a data signal superimposed with a plurality of data values from two or more data sets to be combined into a single data storage without collisions; generate a key for each of the plurality of data values, wherein the respective key is generated via application of a collision-resistant hashing algorithm to an associated data value; execute a first query on a database of the processing server to store each of the plurality of data values and an associated generated key in the database of the processing system as a plurality of key-value pairs; receive a data signal superimposed with a plurality of additional key-value pairs; execute a second query on the database of the processing server to store each of the plurality of additional key-value pairs in the database of the processing server, wherein each key included in the plurality of additional key-value pairs is generated via application of the collision-resistant hashing algorithm to the associated value; receive from an external system, via a communication network, a data signal superimposed with a rehash request that indicates a

hashing algorithm different from the collision-resistant hashing algorithm; and generate, without requiring exchanging the data values or associated keys, a new key for each of the plurality of data values and each value included in the plurality of additional key-value pairs by rehashing the data values using the hashing algorithm received from the external system.

BRIEF DESCRIPTION OF THE DRAWING FIGURES

The scope of the present disclosure is best understood from the following detailed description of exemplary embodiments when read in conjunction with the accompanying drawings. Included in the drawings are the following figures:

WHAT IS CLAIMED IS:

1. A method for generating a universal distributed data storage, comprising:

receiving, by a processing server, a data signal superimposed with a plurality of data values from two or more data sets to be combined into a single data storage without collisions;

generating, by the processing server, a key for each of the plurality of data values, wherein the respective key is generated via application of a collision-resistant hashing algorithm to an associated data value;

executing, by the processing server, a first query on a database of the processing server to store each of the plurality of data values and an associated generated key in the database of the processing server as a plurality of key-value pairs;

receiving, by the processing server, a data signal superimposed with a plurality of additional key-value pairs;

executing, by the processing server, a second query on the database of the processing server to store each of the plurality of additional key-value pairs in the database of the processing server,

wherein each key included in the plurality of additional key-value pairs is generated via application of the collision-resistant hashing algorithm to the associated data value;

receiving, by the processing server, from an external system via a communication network, a data signal superimposed with a rehash request that indicates a hashing algorithm different from the collision-resistant hashing algorithm; and

without requiring exchanging the data values or associated keys, generating, by the processing server, a new key for each of the plurality of data values and each value included in the plurality of additional key-value pairs by rehashing the data values using the hashing algorithm received from the external system.

2. The method of claim 1, wherein the collision-resistant hashing algorithm is an SHA-256 algorithm.

3. The method of claim 1, further comprising:

executing, by the processing server, a third query on the database of the processing server to sort the plurality of key-value pairs and plurality of additional key-value pairs based on the respective key.

4. The method of claim 3, further comprising:

executing, by the processing server, a fourth query on the database of the processing server to replace the key in each of the plurality of key-value pairs and plurality of additional key-value pairs with the respective generated new key.

5. The method of claim 1, wherein the different hashing algorithm is a collision-resistant hashing algorithm.

6. The method of claim 5, wherein the collision-resistant hashing algorithm of the different hashing algorithm is an SHA-256 algorithm.

7. The method of claim 1, wherein the collision-resistant hashing algorithm is such that each generated key is unique with respect to all other generated keys.

8. The method of claim 1, wherein the database of the processing server is a key-value database, and the generated key is a key in the key-value pair and the data value is a value in the key-value pair.

9. A system for generating a universal distributed data storage, comprising:

a database of a processing server;

a processing server configured to:

receive a data signal superimposed with a plurality of data values from two or more data sets to be combined into a single data storage without collisions;

generate a key for each of the plurality of data values, wherein the respective key is generated via application of a collision-resistant hashing algorithm to an associated data value;

execute a first query on a database of the processing server to store each of the plurality of data values and an associated generated key in the database of the processing server as a plurality of key-value pairs;

receive a data signal superimposed with a plurality of additional key-value pairs;

execute a second query on the database of the processing server to store each of the plurality of additional key-value pairs in the database of the processing server, wherein each key included in the plurality of additional key-value pairs is generated via application of the collision-resistant hashing algorithm to the associated value;

receive from an external system, via a communication network, a data signal superimposed with a rehash request that indicates a hashing algorithm different from the collision-resistant hashing algorithm; and

generate, without requiring exchanging the data values or associated keys, a new key for each of the plurality of data values and each value included in the plurality of additional key-value pairs by rehashing the data values using the hashing algorithm received from the external system.

10. The system of claim 9, wherein the collision-resistant hashing algorithm is an SHA-256 algorithm.

11. The system of claim 9, wherein the processing server is further configured to:

execute a third query on the database of the processing server to sort the plurality of key-value pairs and plurality of additional key-value pairs based on the respective key.

12. The system of claim 11, wherein the processing server is further configured to:

execute a fourth query on the database of the processing server to replace the key in each of the plurality of key-value pairs and plurality of additional key-value pairs with the respective generated new key.

13. The system of claim 9, wherein the different hashing algorithm is a collision-resistant hashing algorithm.

14. The system of claim 13, wherein the collision-resistant hashing algorithm of the different hashing algorithm is an SHA-256 algorithm.

15. The system of claim 9, wherein the collision-resistant hashing algorithm is such that each generated key is unique with respect to all other generated keys.

16. The system of claim 9, wherein the database of the processing server is a key-value database, and the generated key is a key in the key-value pair and the data value is a value in the key-value pair.