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.

In a first aspect, the present disclosure provides a method for generating a universal distributed data storage includes: receiving, by a receiving device of a processing server, a data signal superimposed with a plurality of data values; generating, by a hashing module of 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 the associated data value; executing, by a querying module of the processing server, a first query on a database

of the processing server to store each of the plurality of data values and the associated generated key as a plurality of key-value pairs; receiving, by the receiving device of the processing server, a data signal superimposed with a plurality of additional key-value pairs, wherein each value of the plurality of additional key-value pairs is not included in the plurality of data values; and executing, by the querying module of the processing server, a second query on the database of the processing server to store each of the plurality of additional key-value pairs, 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 and is not included in the keys generated for each of the plurality of data values, the method further comprising: receiving, by the receiving device of the processing server, a data signal superimposed with a rehash request, wherein the rehash request indicates a different hashing algorithm; generating, by the hashing module of 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; and executing, by the querying module of the processing server, a third 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.

In a second aspect, the present disclosure provides a system for generating a universal distributed data storage includes: a database of a processing server; a receiving device of the processing server configured to receive a data signal superimposed with a plurality of data values; a hashing module of the processing server configured to 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 the associated data value; and a querying module of the processing server configured to execute a first query on a database of the processing server to store each of the plurality of data values and the associated generated key as a plurality of key-value pairs, wherein the receiving device of the processing server is further configured to receive a data signal superimposed with a plurality of additional key-value pairs, wherein each value of the plurality of additional key-value pairs is not included in the plurality of data values, the querying module of the processing server is further configured to execute a second query on the database of the processing

server to store each of the plurality of additional key-value pairs, and 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 and is not included in the keys generated for each of the plurality of data values; and wherein the receiving device of the processing server is further configured to receive a data signal superimposed with a rehash request, wherein the rehash request indicates a different hashing algorithm, the hashing module of the processing server is further configured to generate a new key for each of the plurality of data values and each value included in the plurality of additional key-value pairs, and the querying module of the processing server is further configured to execute a third 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.

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 receiving device of a processing server, a data signal superimposed with a plurality of data values;

generating, by a hashing module of 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 the associated data value;

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

receiving, by the receiving device of the processing server, a data signal superimposed with a plurality of additional key-value pairs, wherein each value of the plurality of additional key-value pairs is not included in the plurality of data values; and

executing, by the querying module of the processing server, a second query on the database of the processing server to store each of the plurality of additional key-value pairs, 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 and is not included in the keys generated for each of the plurality of data values, the method further comprising:

receiving, by the receiving device of the processing server, a data signal superimposed with a rehash request, wherein the rehash request indicates a different hashing algorithm;

generating, by the hashing module of 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; and

executing, by the querying module of the processing server, a third 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.

~~2. The method of claim 1, wherein the hashing algorithm is a collision-resistant hash algorithm.~~

~~3~~2. The method of claim 2, wherein the collision-resistant hashing algorithm is an SHA-256 algorithm.

~~4~~3. The method of claim 1, further comprising:

executing, by the querying module of 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.

~~5. The method of claim 1, further comprising:~~

~~receiving, by the receiving device of the processing server, a data signal superimposed with a rehash request, wherein the rehash request indicates a different hashing algorithm;~~

~~generating, by the hashing module of 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; and~~

~~executing, by the querying module of the processing server, a third 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.~~

~~6~~4. The method of claim ~~5~~1, wherein the different hashing algorithm is a collision-resistant hashing algorithm.

~~7~~5. The method of claim ~~6~~4, wherein the different hashing algorithm is an SHA-256 algorithm.

~~8~~6. The method of claim ~~5~~1, further comprising:

executing, by the querying module of the processing server, a fourth 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 generated new key.

~~9~~7. The method of claim 1, further comprising:

receiving, by the receiving device of the processing server, a data signal superimposed with a value request, wherein the value request includes at least a requesting key;

executing, by the querying module of the processing server, a third query on the database of the processing server to identify a specific key-value pair where the key corresponds to the requesting key; and

electronically transmitting, by a transmitting device of the processing server, a data signal superimposed with at least the value included in the identified specific key-value pair in response to the received value request.

~~10. The method of claim 1, wherein the hashing algorithm is such that each generated key is unique with respect to all other generated keys.~~

~~11~~8. A system for generating a universal distributed data storage, comprising:

a database of a processing server;

a receiving device of the processing server configured to receive a data signal superimposed with a plurality of data values;

a hashing module of the processing server configured to 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 the associated data value; and

a querying module of the processing server configured to execute a first query on a database of the processing server to store each of the plurality of data values and the associated generated key as a plurality of key-value pairs, wherein

the receiving device of the processing server is further configured to receive a data signal superimposed with a plurality of additional key-value pairs, wherein each value of the plurality of additional key-value pairs is not included in the plurality of data values,

the querying module of the processing server is further configured to execute a second query on the database of the processing server to store each of the plurality of additional key-value pairs, and

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 and is not included in the keys generated for each of the plurality of data values; and wherein

the receiving device of the processing server is further configured to receive a data signal superimposed with a rehash request, wherein the rehash request indicates a different hashing algorithm,

the hashing module of the processing server is further configured to generate a new key for each of the plurality of data values and each value included in the plurality of additional key-value pairs, and

the querying module of the processing server is further configured to execute a third 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.

~~12. The system of claim 11, wherein the hashing algorithm is a collision-resistant hash algorithm.~~

~~13~~9. The system of claim ~~12~~8, wherein the collision-resistant hashing algorithm is an SHA-256 algorithm.

~~14~~10. The system of claim ~~11~~8, wherein the querying module of 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.

~~15. The system of claim 11, wherein~~

~~the receiving device of the processing server is further configured to receive a data signal superimposed with a rehash request, wherein the rehash request indicates a different hashing algorithm,~~

~~the hashing module of the processing server is further configured to generate a new key for each of the plurality of data values and each value included in the plurality of additional key-value pairs, and~~

~~the querying module of the processing server is further configured to execute a third 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.~~

~~16~~11. The system of claim ~~15~~8, wherein the different hashing algorithm is a collision-resistant hashing algorithm.

~~17~~12. The system of claim ~~16~~11, wherein the different hashing algorithm is an SHA-256 algorithm.

~~18~~13. The system of claim ~~15~~8, wherein the querying module of the processing server is further configured to execute a fourth 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 generated new key.

~~19~~14. The system of claim ~~11~~8, further comprising:

a transmitting device of the processing server, wherein

the receiving device of the processing server is further configured to receive a data signal superimposed with a value request, wherein the value request includes at least a requesting key,

the querying module of the processing server is further configured to execute a third query on the database of the processing server to identify a specific key-value pair where the key corresponds to the requesting key, and

the transmitting device of the processing server is configured to electronically transmit a data signal superimposed with at least the value included in the identified specific key-value pair in response to the received value request.

~~20. The system of claim 11, wherein the hashing algorithm is such that each generated key is unique with respect to all other generated keys.~~