KIP-665: Kafka Connect Hash SMT

Status

Current state: *Under Vote*

Discussion thread: *here [Change the link from the KIP proposal email archive to your own email thread]*

JIRA: *KAFKA-10299*

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast).

Motivation

A previous contribution I made to https://github.com/aiven/aiven-kafka-connect-transforms was suggested as by a member of confluent as being a nice addition to the out of the box Kafka Connect SMTs. The discussion is here https://github.com/aiven/aiven-kafka-connect-transforms/issues/9#issuecomment-662378057. The proposed change would add a new Kafka Connect SMT which would allow for keys or values to be hashed using the configured algorithm. The addition of this would allow for sensitive fields to be obfuscated to prevent private information such as ssn or other identifiable information from flowing.

Currently there exists a MaskField SMT but that would completely remove the value by setting it to an equivalent null value. One problem with this would be that you'd not be able to know in the case of say a password going through the mask transform it would become "" which could mean that no password was present in the message, or it was removed. However this hash transformer would remove this ambiguity if that makes sense. The proposed hash functions would be MD5, SHA1, SHA256. which are all supported via MessageDigest.

Public Interfaces

One new class connect/transforms/src/main/java/org/apache/kafka/connect/transforms/Hash.java and a helper class connect/transforms/src/main/java/org/apache/kafka/connect/transforms/util/Hex.java are proposed additions. No modifications required to existing interfaces.

Proposed Changes

The proposed change can be viewed here https://github.com/apache/kafka/pull/9057 it would allow for hashing specific fields within a kafka connect message value, or the entire value, additionally the key could be hashed if desired. The configuration would look something like the folllowing. where type is Either Key or Value.

```
transforms=HashEmail
transforms.HashEmail.type=org.apache.kafka.connect.transforms.Hash$Value
transforms.HashEmail.field.name=email
transforms.HashEmail.function=sha1

Based on feedback from Gunnar Morling (https://debezium.io/documentation/reference/connectors/mysql#mysql-
property-column-mask-hash) I think that this should also support
1) an optional salt, which would be set via transforms.HashEmail.salt
2) a comma separated list of fields where a period is used to denote nested fields
Given these suggestions

transforms=HashFields
transforms.HashFields.type=org.apache.kafka.connect.transforms.Hash$Value
transforms.HashFields.field.name=user.email,user.ssn,contact
transforms.HashFields.function=sha1
transforms.HashFields.salt=F4xJK03Ab
```

Compatibility, Deprecation, and Migration Plan

- NA

Rejected Alternatives

NA