Google

Private Join and Compute

Problem statement

A city is evaluating whether to offer a weekend train service to an up-and-coming neighborhood and will base its decision on how much train riders spend at local businesses in that neighborhood.

Challenge

- The city knows who rides the train
- Local businesses in the neighborhood use a point-of-sale provider that knows the value of consumer transactions
- The city and point-of-sale provider do not want to share personally identifiable information with each other
- How can the city work together with local merchants to answer this question without revealing any personally identifiable information about train riders, where they travel, or what they buy?

1 Each party has its own data

City and businesses/point-of-sale data

2 Encrypting each party's data

First, both parties encrypt their data with private keys so that it's not accessible or decipherable to anyone else.

3 Exchanging encrypted data

Then, each party sends its encrypted data to the other party.

4 Double encrypting

Both parties encrypt with their own private keys, resulting in double-encrypted data.

The double-encrypted IDs can be compared but can't be decrypted by either party individually.

5 Finding intersections

The point-of-sale provider can send the double-encrypted train rider data back to the city in shuffled order.

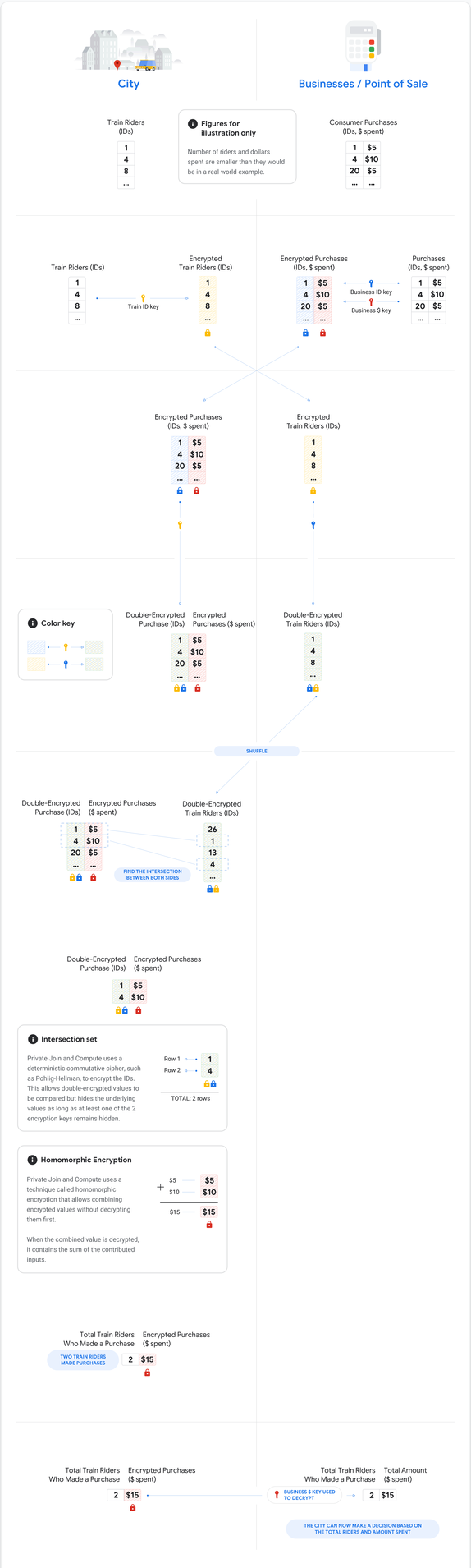
The city can now compare the IDs to find the overlapping set of users but won't know anything about the value of their purchases at local businesses.

6 Joining encrypted values

By counting intersecting rows, the city can now calculate the value of the purchases made by train riders.

7 Aggregating

With homomorphic encryption, encrypted purchase values can be summed, but the actual amount spent by train riders can't be viewed.

8 Summarizing, decrypting, and viewing

The city can pass its encrypted purchase data to the point-of-sale provider, where the data can be decrypted and the total amount spent can be viewed.