**Use Case: Private to Private – Domestic Trusted Group**

|  |  |  |
| --- | --- | --- |
| **Medium trust** | **Encryption in transit, operation, rest** | **Results summary available to all parties** |

**Overview**

This use case involves comparing Suspicious Person Lists (SPLs) between two trusted domestic private parties. The objective is to identify common suspicious individuals while maintaining a high level of data privacy and security. The results of the comparison are shared with all involved parties in the form of a summary, with full details available only by mutual agreement.

**Privacy and Trust**

In this scenario, a medium level of trust is established between the parties involved. To ensure data privacy and security, encryption is utilized at every stage of data handling:

* **In Transit**: Data is encrypted during transmission to prevent unauthorized access.
* **In Operation**: Data is processed in its encrypted form to maintain confidentiality.
* **At Rest**: Data is stored in an encrypted state to protect it from unauthorized access.

The comparison process focuses solely on matching personal information of suspicious customers that is common between the two parties. This approach minimizes the amount of personal data shared and ensures that only the necessary information is disclosed.

**Governance and Compliance**

To ensure effective management of this use case, it is critical to implement robust governance frameworks among the parties involved. Considering the sensitive characteristics of the data, compliance with privacy regulations and applicable legislation is paramount. This could require the modification of privacy policies and terms of service to acquire customer consent for the proposed data sharing. The parties are unlikely to depend on current exceptions in privacy laws concerning law enforcement, which highlights the necessity of establishing clear agreements and consent mechanisms.

**Technologies**

Homomorphic Encryption: Homomorphic Encryption is a sophisticated encryption method that enables operations to be conducted on encrypted data without requiring decryption. In the realm of SPL comparison, this technique can facilitate secure comparisons while maintaining the encryption of the data. Consequently, it guarantees that sensitive information is safeguarded throughout the comparison procedure.

Symmetric encryption: The process of symmetric encryption is utilized to safeguard the summary of the comparison results before it is distributed to all relevant parties. This approach ensures that the summary remains confidential and is only available to authorized individuals. Furthermore, symmetric encryption is applied to the detailed results, which will be shared only upon mutual agreement between the parties involved.

**Code Explanation**

**Required Packages**

* The required packages (phe, pycryptodome, and faker) are installed and imported for the code to function correctly.

**Synthetic Data Generation**

* The generate\_synthetic\_spl function generates a synthetic dataset for SPL using the Faker library. It creates entries with customer information, including customer\_id, name, dob, ssn, address, transaction\_activity, and risk\_score.
* The generated data simulates sensitive information used in the SPL comparison.

**Generate SPL Data for Two Institutions**

* Two synthetic SPL datasets (spl\_institution1 and spl\_institution2) are created, each containing 5000 entries.

**Homomorphic Encryption (HE) - Paillier Key Generation**

* Paillier public and private keys are generated using paillier.generate\_paillier\_keypair().
* The public key is used to encrypt the data, ensuring that data remains confidential during processing.

**Encryption Function**

* encrypt\_data: This function encrypts a list of data using the provided public key.
* It is applied to the risk\_score column of both SPL datasets.

**Encrypting SPL Data - Risk Scores**

* The risk\_score values from both institutions' SPLs are encrypted using the public key, ensuring that the data is processed in an encrypted form.

**Encrypted Data Comparison**

* compare\_encrypted\_data: This function compares the encrypted risk\_score values between the two institutions to find matches.
* Matches are identified without decrypting the data, maintaining privacy throughout the process.

**Performing Comparison on Encrypted Data**

* The comparison is executed, and the count of matched items is printed.
* The comparison results indicate how many risk\_score values are common between the two institutions, preserving privacy since the actual scores remain encrypted.