隐私保护交互协议，主要用于在两个参与方（P1 和 P2）之间进行数据匹配与计算，同时保护双方的隐私数据不被泄露。协议通过哈希函数、模运算、同态加密等技术，实现了“在不暴露具体数据的情况下，计算双方共同拥有的密码对应的数值总和”的目标。

假设 P1 拥有一组密码集合，P2 拥有一组“密码 - 数值”对集合。双方希望在不泄露各自具体密码的前提下，找到两集合中共同存在的密码，并计算这些共同密码对应的数值总和。协议依赖以下技术保障隐私与功能：

1. **哈希函数**：使用 SHA-256 对密码进行哈希，将原始密码转换为不可逆的哈希值，避免原始密码泄露；
2. **模运算与大素数**：基于大素数的模指数运算，用于“盲化”哈希值（掩盖原始哈希结果），防止对方直接识别匹配的密码；
3. **Paillier 同态加密**：一种支持“加密状态下加法运算”的加密算法，允许 P2 对数值加密后，P1 在加密状态下对匹配的数值求和，最终由 P2 解密得到结果（避免中间过程数值泄露）；
4. **随机化处理**：通过 random.shuffle 打乱数据顺序，进一步隐藏数据关联。

协议执行步骤

初始化阶段：

参与方 P_2生成 Paillier 加密算法的公钥与私钥，并将公钥共享给参与方 P_1。

第一阶段交互（P_1到 P_2）：

1. P_1对其数据集 V 中的每个标识 v 执行哈希运算，得到 H (v)
2. 使用自身持有的秘密随机数 k_1对哈希结果进行盲化处理：v_k1 = H (v)^k_1 mod p
3. 将所有盲化后的值随机打乱顺序，形成集合 V' 并发送给 P_2

第二阶段交互（P_2到 P_1）：

1. P_2接收 P_1发送的 V' 后，使用自身秘密随机数 k_2进行二次盲化，生成集合 Z = {v_k1^k_2 mod p | 所有 v_k1 ∈ V'}
2. 对自身数据集 W 中的每个标识 w 执行：

- 计算哈希值 H (w) 并进行盲化处理：$w\_k2 = H(w)^{k_2} \bmod p$
- 使用 P_1提供的 Paillier 公钥加密与 w 关联的风险值 t，得到 encrypted_t

3. 将所有 (w_k2, encrypted_t) 对与集合 Z 共同随机打乱顺序后，一并发送给 P_1

第三阶段交互（P_1到 P_2）：

1. P_1接收 P_2发送的数据后，对每个 w_k2 执行：
   - 使用自身秘密随机数 k_1计算：$w\_k1k2 = w\_k2^{k_1} \bmod p$
   - 检查 w_k1k2 是否存在于集合 Z 中，若存在则判定为交集元素
2. 收集所有交集元素对应的 encrypted_t
3. 利用 Paillier 加密的加法同态特性，对收集到的 encrypted_t 进行累加运算，得到加密总和 encrypted_sum
4. 将 encrypted_sum 发送给 P_2

最终解密阶段（P_2执行）：

P_2使用自身持有的 Paillier 私钥对 encrypted_sum 进行解密操作，得到交集元素对应的风险总值。

```python
# ------------------ 参与方P1：准备本地数据集 ------------------
local_identifiers = ["password1", "123456", "qwerty"]
secret_k1 = random.randint( a: 2,  b: 1000)  # P1的秘密随机数
mod_prime = 2 ** 521 - 1  # 用于模运算的大素数


2个用法
def identifier_hash(input_str):
    """将字符串标识符转换为哈希整数"""
    return int(hashlib.sha256(input_str.encode()).hexdigest(), 16)


# P1对本地标识符进行盲化处理
p1_blinded = [pow(identifier_hash(v), secret_k1, mod_prime) for v in local_identifiers]
random.shuffle(p1_blinded)  # 打乱顺序增强隐私

# ------------------ 参与方P2：处理本地数据并返回 ------------------
remote_identifiers = ["123456", "letmein", "password2"]
risk_scores = [5, 10, 7]  # 对应每个标识符的风险分数
secret_k2 = random.randint( a: 2,  b: 1000)  # P2的秘密随机数
```

运行结果

```
C:\Users\swj\AppData\Local\Programs\Python\Python312\python.exe C:\Users\swj\project6\test.py
交集标识符的总风险值： 5
```