基于公钥密码的网恋匹配系统

项目简介与思路

本项目旨在设计并实现一个基于公钥加密的校内网恋匹配系统。设想如下场景：存在一个校内网恋匹配平台，用户可以注册并填写可公开展示的个人信息，如姓名、性别、年龄、身高、体重、学历、职业、爱好等。平台会每日向用户推送若干匹配对象，推送为双向的：若向 Alice 推送了 Bob，则 Bob 也会被推送 Alice。用户对每一位推送对象可以选择“接受”或“拒绝”，仅当双方都选择“接受”时，匹配才成立。

匹配逐个推送，用户在完成当前对象的选择（接受/拒绝）后，才会收到下一位推送对象的资料。匹配不要求两人同时在线，只要双方在上线后分别完成选择，平台即可尝试匹配。

该场景面临以下两个安全问题：

- **平台隐私泄露问题**：传统实现中，平台可以获知用户的选择及匹配结果。由于平台维护者可能为校内人员，用户并不希望自己的选择或匹配结果被平台知晓。
- **单方选择泄露问题**：若采用常规加密通信方式进行匹配，用户可以得知对方的选择，造成心理尴尬。例如 Bob 想与 Alice 匹配，而 Alice 拒绝了 Bob，这种情况发生时，Bob并不希望 Alice 知道自己曾选择了她。

因此，项目目标是在平台不主动作恶，仅可能偷窥用户选择的前提下，设计一个系统满足以下两点：

- 平台无法得知用户的选择及最终匹配结果。
- 若一方拒绝匹配，则无法得知对方的选择。例如，若 Alice 拒绝了 Bob，则 Alice 无法知道 Bob 是否选择了自己。

我们可以将匹配操作建模为逻辑与（AND）运算：

- “接受” 表示输入为 1，“拒绝” 表示输入为 0。
- 匹配结果即为双方输入的逻辑与结果（1 表示匹配成功，0 表示失败）：
    - 如果Alice输入1，Bob输入1，则最终结果为1，即匹配成功。
    - 如果Alice输入1，Bob输入0，则最终结果为0，即匹配失败。
    - 如果Alice输入0，Bob输入1，则最终结果为0，即匹配失败。
    - 如果Alice输入0，Bob输入0，则最终结果为0，即匹配失败。

相当于我们希望两方获得最终结果，但无法知道中间的计算，即**无法推断超出结果的信息**。比如，如果 Alice 输入0，Bob 输入1，则最终结果为0，即匹配失败。但是 Alice 无法知道 Bob 的输入为 1 还是 0，即 Bob 是否选择了自己，因为从最终结果 0，无法推断出对方的选择。而Bob知道最终结果为0，自然可以推断出对方拒绝了自己（故仍符合**无法推断超出结果的信息**）。我们的要求为双方无法得到超出计算结果能够推断的信息。

那么，应该如何设计这么一个系统呢？

以下提供一种参考设计思路：

首先，可以使用 Diffie-Hellman 密钥交换机制解决第一个问题。每位用户选择一个私钥 x，计算 $y = g^x$ 后上传至平台。例如，Alice 选择 x_A，上传 $y_A = g^{x_A}$；Bob 选择 x_B，上传 $y_B = g^{x_B}$。如果平台推送两方进行匹配，那么Alice可以收到Bob的 y_B，Bob 也可以收到 Alice 的 y_A，Alice和Bob可以计算共享密钥 $k = g^{x_A \cdot x_B}$，后续双方的匹配选择，就可以借助 k 进行加密。

为解决第二个问题，我们可以利用 ElGamal 加密方案的同态性：

我们先回顾一下ElGamal加密方案（假设公共参数已经事先确定）：

> - 公共参数：$(\mathbb{G}, g, q)$，其中 $\mathbb{G}$ 是循环群，g 为生成元，q 为群的阶。
> - 密钥对：(pk, sk)，其中 $pk = h = g^x$，$sk = x$。
> - 加密：输入明文 $M \in \mathbb{G}$，随机选取 $k \in \mathbb{Z}_q$，输出密文 $(C_1, C_2) = (g^k, M \cdot h^k)$。
> - 解密：$M = C_2 \cdot C_1^{-x}$。
> - 同态性质：
>   - 相乘：$\mathrm{Enc}(M_1) \cdot \mathrm{Enc}(M_2) = \mathrm{Enc}(M_1 \cdot M_2) = (g^{k_1+k_2}, M_1 \cdot M_2 \cdot h^{k_1+k_2})$
>   - 幂运算：$\mathrm{Enc}(M)^r = \mathrm{Enc}(M^r) = (g^{k \cdot r}, M^r \cdot h^{k \cdot r})$

双方可通过哈希函数等密钥派生函数 H 从共享密钥 k 派生出 $x = H(k) \in \mathbb{Z}_q$，作为用于 ElGamal 加密的共享私钥（此私钥为两方共同持有）。

用户将自己的匹配选择加密后提交平台：

- 若选择接受，对明文 $M = 1$ 进行加密。
- 若选择拒绝，对随机选取的 $M \in \mathbb{G}$ 进行加密。

平台在接收到双方密文后，进行密文同态乘法运算。若双方均选择接受，则密文为加密的 1，匹配成功；若任一方拒绝，则结果为群中随机元素，匹配失败。

也就是，平台在不知道双方选择的情况下，可以进行密文上的匹配运算。如果双方得到匹配后的运算结果，因为双方都持有私钥，那么双方都可以进行解密，得到最后的匹配结果（1或者随机元素）。

值得注意的是，进行简单的密文上的乘法是不够的，假设 Alice 提交的密文是 C_A，在获得两个密文相乘的结果密文 $C_\times = C_A \cdot C_B$ 后，可以反推出Bob的密文 C_B，获得对方的加密选择，并解密。

为防止单方根据最终密文推断出对方的选择，平台需对同态计算后的密文进行再随机化：对密文做一次幂运算。例如，选择随机数 r，将密文对应的明文 M 变换为 M^r。若 $M = 1$，则 $M^r = 1$，保持不变；若 M 为随机值，则 M^r 仍为随机元素，不可用于推断原值。

最终，匹配双方可解密密文获取匹配结果：

- 解密为 1：匹配成功。
- 解密为随机元素：匹配失败。

所以，以上思路可以同时解决上述两个问题。

进一步思考：是否可以在匹配成功后，使双方自动交换联系方式等非公开信息？若匹配失败，则无法获取任何信息？

一种扩展思路如下：

用户除了上传加密匹配选择外，还上传一个用 ElGamal 加密的密码（如用于解密联系方式的对称密钥，或者某个带加密功能文件的口令），记为 C_m，以及一个用该密码加密的联系方式密文/文件。平台对匹配结果密文与 C_m 进行同态乘法：

- 匹配成功（密文对应明文为 1）：结果仍为加密的原始密钥，双方可解密后获取联系方式；
- 匹配失败（密文为随机元素）：结果为无效密钥，无法解密获取联系方式。

任务要求

- 编写程序实现上述系统，编程语言不限。**最低要求为本地模拟完成匹配结果计算过程**（不含联系方式交换）。鼓励开发完整系统。思路仅为参考思路，若有其他思路，可以提出新方案或者进一步拓展和完善。
- 进行程序开发，完成上述系统，不限制变成语言。最低要求是完成匹配结果（不需要包含联系方式交换）的计算过程进行本地模拟。鼓励按照上述项目需求开发一个完整的系统。思路仅为参考思路，若有其他思路，可以进一步拓展和完善。
- 可自由使用任意算法库，亦可复用上一项目的相关代码。
- 鼓励在效率与系统鲁棒性方面进行优化。

提交材料

1. **代码**：完整、可运行的源代码，需包含清晰的注释，解释关键代码段的逻辑与功能。
2. **项目报告**：
    - **需求分析**：分析项目需求，明确项目的目标。
    - **实现方案**：详细说明所采用的技术路线和各算法模块的作用。
    - **项目特色**：说明独特设计、性能优化点及创新内容。
    - **核心代码解释**：选取关键代码进行解释，便于读者理解实现思路。
    - **编译运行说明**：如果代码的编译和运行过程比较复杂，需要附上详细的编译和运行步骤，包括所需的环境配置、依赖安装以及命令行指令等，确保能够顺利运行代码。
    - **运行结果展示**：提供程序运行结果的展示，可以是运行截图、输出日志等。
    - **困难与解决思路**：记录在实现代码过程中遇到的困难和挑战，以及相应的解决思路和方案。

请注意

1. 请在截止日期前提交**项目报告**与**程序**。截止日期是**7月12日23：59。超过截止日期的提交将无效**。请在截止日期前发送至邮箱i@liuyi.pro。
2. 请分开提交项目报告与源代码（即不要放在同个压缩包下）：报告与源代码命名方式为：姓名-学号-网恋匹配项目，报告为pdf，源代码请传zip压缩包。
3. 你的分数还将取决于你的源代码和报告的质量。你的报告应该容易理解，并很好地描述你的工作，特别是你工作的亮点。

4. 请更加注意您的代码风格。您有足够的时间来编写具有正确结果和良好代码风格的代码。如果你的代码风格很糟糕，可能会被扣分。可以参考Google C++风格指南(http://google.github.io/styleguide/cppguide.html)或其他一些代码风格指南。