项目概述

本项目是一个基于 Flask 的 Web 应用，专为 **联邦学习模型参数的共享与安全检索** 设计。应用中使用 **密码属性加密（CP-ABE）** 实现了用户认证、元数据管理和加密检索功能。此应用允许用户在联邦学习环境中：

- 创建参数共享空间
- 上传带有元数据的模型参数文件
- 基于关键词对共享的模型参数文件进行安全检索

通过模块化设计，系统具备良好的扩展性和可维护性，支持复杂的访问控制需求。

目录结构与模块说明

1. Main.py

主程序入口，负责创建 Flask 应用实例和初始化联邦学习参数存储空间。启动应用时，会确保参数存储目录的存在，并以调试模式运行，方便开发时查看实时反馈。

2. __init__.py

此文件用于初始化 Flask 应用，集成了 SQLAlchemy 以管理用户和元数据，使用 Flask-Login 进行用户会话管理。该模块还负责数据库初始化、参数存储目录创建，并注册视图和认证相关的蓝图。

3. auth.py

管理用户认证和会话的模块。定义了用户登录和退出的路由，提供 create_users() 函数初始化用户，仅在数据库为空时创建若干测试用户。登录路由检查用户凭证，并管理用户会话状态。

4. models.py

定义了用户和模型参数的数据库模型，用户模型包含用户 ID、用户名、密码和角色属性等字段。模型参数元数据模型则包括参数文件的 ID、描述、关键词、上传时间等信息。SQLAlchemy 用于与数据库交互，支持基于用户角色的权限管理。

5. view.py

提供应用主要功能的路由，包括用户欢迎页、参数检索、文件读取、参数文件上传和空间创建等。每个路由都带有访问控制，确保只有经过身份验证的用户可以访问相关功能。

6. fm.py

提供联邦学习参数文件的管理功能，包括使用 CP-ABE 对文件加密和解密、元数据的加载与维护、关键词检索等实用功能。通过管理元数据文件及关键字索引，实现对共享模型参数的高效、安全检索。

7. KeyGen.py

生成 CP-ABE 加密方案的加密密钥脚本。设定密钥对，生成公共密钥和主密钥，并根据用户角色生成私密密钥。密钥文件的原子性保存确保应用能够基于用户角色对模型参数进行加密和解密。

主要功能

- **用户管理**：支持用户登录、登出和密码管理，密码通过哈希存储确保安全性。
- **模型参数管理**：支持用户上传、加密、解密和下载模型参数文件，并为每个文件维护可检索的元数据。
- **元数据处理**：管理联邦学习模型参数的元数据结构，支持文件的高效检索与组织。

- **加密与解密**：参数文件存储前加密，用户请求时按角色安全解密。

关键技术组件

- **Flask 框架**：作为 Web 框架，构建整个应用的逻辑结构。
- **SQLAlchemy**：用于数据库交互，管理用户数据和模型参数元数据。
- **Flask-Login**：用于用户会话管理，确保只有已认证用户能访问特定路由。
- **CP-ABE 加密**：提供文件加密能力，实现基于用户角色的安全访问控制。

安装与使用

1. **安装依赖**：确保已安装 Flask、Flask-SQLAlchemy、Flask-Login 和 Charm-Crypto 等库。

```
pip install flask flask_sqlalchemy flask_login charm-crypto
```

2. **运行应用**： 执行 main.py 启动应用。这将初始化必要的数据库和用户结构。

```
python main.py
```

3. **访问应用**： 在浏览器中访问 http://localhost:5000 以查看应用界面。

安全功能

- **密码哈希**：用户密码通过哈希存储，即便数据库泄露，用户凭证也能得到保护。
- **会话管理**：使用 Flask-Login 管理用户会话，确保只有已认证用户可以访问敏感路由。
- **角色访问控制**：通过 CP-ABE 加密策略，基于用户角色进行数据访问权限控制。