**软件开发文档1.0.0**

**网络通信软件开发文档**

**文档版本**

* 版本号：1.0.0
* 更新日期：2025年4月15日

**目录**

1. **系统概述**
2. **系统架构**
3. **模块详细说明**
4. **数据库设计**
5. **通信协议**
6. **安全机制**
7. **安装与配置指南**
8. **使用说明**
9. **测试方法**
10. **常见问题**

**1. 系统概述**

本系统是一个基于TCP协议的加密通信软件，支持多用户实时聊天、文件传输、离线消息存储及管理员功能。核心功能包括：

* **用户认证**：注册、登录、密码哈希存储
* **实时通信**：文本消息、文件传输
* **离线消息管理**：自动存储和转发离线消息
* **管理员功能**：用户管理、公告发送
* **安全机制**：SSL/TLS加密通信、密码哈希存储（Bcrypt）

**2. 系统架构**

**整体架构图**

|  |
| --- |
| Plain Text +----------------+ +----------------+  | 客户端 | <------> | 服务器 |  +----------------+ +----------------+   ↑ ↑   | SSL加密 | SSL加密   | |  +----------------+ +----------------+  | GUI界面 | | 数据库(SQLite) |  +----------------+ +----------------+ |

**模块交互流程**

1. 客户端通过SSL连接服务器
2. 用户完成注册/登录认证
3. 服务器验证用户并同步离线消息
4. 客户端发送消息/文件，服务器转发或存储
5. 管理员执行用户管理或发送公告

**3. 模块详细说明**

**3.1 SSL证书模块（SSL.py）**

* **功能**：生成自签名证书，支持多域名和电子邮件地址
* **输入参数**：
* domain\_list：域名列表（首个作为CN）
* email：证书关联邮箱
* **输出文件**：
* .crt（证书文件）、.key（私钥文件）、.pem（合并文件）
* **依赖库**：OpenSSL

**3.2 数据库模块（database.py）**

* **表结构**：
* **users表**：

|  |
| --- |
| SQL CREATE TABLE users (  id INTEGER PRIMARY KEY,  username TEXT UNIQUE NOT NULL,  password\_hash TEXT NOT NULL,  is\_admin BOOLEAN DEFAULT FALSE,  created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ) |

* **offline\_messages表**：

|  |
| --- |
| SQL CREATE TABLE offline\_messages (  id INTEGER PRIMARY KEY,  sender TEXT NOT NULL,  receiver TEXT NOT NULL,  message\_type TEXT NOT NULL, -- 'chat' 或 'file'  content BLOB NOT NULL,  filename TEXT,  timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ) |

* **关键方法**：
* add\_user()：添加用户（密码哈希存储）
* save\_offline\_message()：保存离线消息
* get\_offline\_messages()：获取并删除离线消息

**3.3 客户端模块（gui.py）**

* **界面组件**：
* 登录/注册界面
* 主聊天界面（用户列表、消息框、文件发送按钮）
* 管理员面板（用户管理、公告发送）
* **功能逻辑**：
* 使用SSL连接服务器
* 发送消息/文件时选择目标用户
* 自动接收实时消息和历史离线消息

**3.4 协议模块（protocol.py）**

* **消息格式**：
* **头部**：JSON格式，包含type（类型）、length（消息体长度）、附加字段（如filename）
* **消息体**：文本（UTF-8编码）或二进制数据
* **关键方法**：
* send\_message()：发送消息（头部长度固定4字节）
* recv\_message()：接收消息（按头部长度解析）

**3.5 服务器模块（server.py）**

* **核心功能**：
* 多线程处理客户端连接
* 用户认证（注册/登录）
* 消息转发（在线用户）或存储（离线用户）
* 管理员命令处理（list\_users、delete\_user、announcement）
* **线程安全**：使用client\_map\_lock确保客户端映射的线程安全

**4. 数据库设计**

**ER图**

|  |
| --- |
| Plain Text +-------------+ +-------------------+  | users | | offline\_messages |  +-------------+ +-------------------+  | id |<----->| sender |  | username | | receiver |  | password\_hash| | message\_type |  | is\_admin | | content |  | created\_at | | filename |  +-------------+ | timestamp |   +-------------------+ |

**5. 通信协议**

**消息类型**

|  |  |  |
| --- | --- | --- |
| 类型 | 描述 | 附加字段示例 |
| chat | 文本消息 | {"to": "目标用户"} |
| file | 文件传输 | {"filename": "文件名称"} |
| admin\_command | 管理员命令 | {"action": "list\_users"} |
| error | 错误反馈 | {"reason": "错误原因"} |

**6. 安全机制**

* **SSL/TLS加密**：客户端与服务器通信全程加密
* **密码哈希**：使用Bcrypt生成并存储密码哈希
* **证书验证**：客户端验证服务器证书有效性

**7. 安装与配置指南**

**依赖安装**

|  |
| --- |
| Bash pip install pyopenssl sqlite3 bcrypt |

**证书生成**

|  |
| --- |
| Bash python SSL.py # 生成tsetcn.crt、tsetcn.key、tsetcn.pem |

**服务器启动**

|  |
| --- |
| Bash python server.py |

**客户端启动**

|  |
| --- |
| Bash python gui.py |

**8. 使用说明**

1. **用户注册/登录**：选择登录或注册，输入用户名和密码
2. **发送消息**：选择在线用户，输入内容后点击“发送”
3. **发送文件**：点击“发送文件”按钮选择本地文件
4. **管理员功能**：登录管理员账户后，通过“管理面板”操作

**9. 测试方法**

* **单元测试**：验证数据库操作、消息协议解析
* **集成测试**：模拟多用户同时在线通信
* **压力测试**：使用JMeter测试高并发场景

**10. 常见问题**

1. **证书错误**：确保客户端证书路径正确（默认SSL/tsetcn.crt）
2. **用户重复注册**：数据库username字段为唯一约束
3. **文件保存失败**：检查files目录权限

**备注**：无