SecurePaste - Encrypted Text Sharing Tool

A secure text sharing tool with end-to-end encryption, customizable expiration options, and self-destruct capabilities for integration with 2lecybersec.com.

Features

- **End-to-End Encryption**: All content is encrypted in the browser before being sent to the server using AES-256-GCM
- **Customizable Expiration**: Multiple expiration options (1 hour, 1 day, 1 week, 1 month, 3 months, 6 months, 1 year, 5 years, never expire, custom)
- **Self-Destruct**: Option to automatically delete content after the first view
- **Admin Integration**: API endpoints for administration and monitoring
- **Responsive Design**: Matches 2lecybersec.com visual style and works on all devices
- **Secure by Design**: The server never has access to unencrypted content

Technical Implementation

Frontend

- Responsive HTML/CSS layout matching 2lecybersec.com style
- Client-side encryption using Web Crypto API
- JavaScript for handling encryption, decryption, and API communication

Backend

- Flask application with SQLAlchemy for database operations
- MySQL database for storing encrypted content
- RESTful API endpoints for paste creation, retrieval, and management
- Automatic expiration and cleanup mechanisms

Integration Guide

Installation

1. Clone the repository to your server

2. Set up a virtual environment and install dependencies: `python -m venv venv source venv/bin/activate pip install -r requirements.txt`
3. Configure the database connection in `src/main.py`
4. Run the application: `python -m src.main`

Integration with 2lecybersec.com

To integrate this tool with the 2lecybersec.com website:

1. **Frontend Integration**:
2. Copy the contents of `src/static/` to your web server

3. Update the API endpoints in `secure-paste.js` to point to your backend server

4. **Backend Integration**:

5. Deploy the Flask application on your server
6. Configure the database connection

7. Set up proper authentication for admin endpoints

8. **Admin Integration**:

9. Use the admin API endpoints to manage pastes:
    - `GET /api/admin/pastes` - List all pastes (with pagination)
    - `POST /api/admin/cleanup` - Clean up expired pastes
    - `DELETE /api/pastes/<paste_id>` - Delete a specific paste

Security Considerations

- The encryption key is never sent to the server, it remains in the URL fragment
- All content is encrypted client-side before transmission
- The server only stores encrypted data
- Self-destruct pastes are permanently deleted after viewing
- Expired pastes are automatically cleaned up

Customization

You can customize the appearance by modifying the CSS in the HTML file to match any updates to the 2lecybersec.com design.

License

This project is proprietary and intended for exclusive use by 2lecybersec.com.