

Encrypt Wall

Securing Social Media Applications

Michael McGuinness | 16322635

Git Repo: <https://github.com/DaVinciTachyon/EncryptWall>

Assignment Brief

The aim of this project is to develop a secure social media application for Facebook, Twitter, WhatsApp, etc., or for your own social networking app. For example, your application will secure the Facebook Wall, such that only people that are part of your "Secure Facebook Group" will be able to decrypt each other's posts. To all other users of the system the post will appear as ciphertext.

You are required to design and implement a suitable key management system for your application that allows any member of the group to share social media messages securely, and allows you to add or remove people from a group. You are free to implement your application for a desktop or mobile platform and make use of any open source cryptographic libraries.

Solution

I understood the task to require the creation of user accounts that can be members of groups. Each of the users has a wall which allows them to see all of the posts, however will decrypt only the ones that are directed to a group they are a member of. Users may be members of multiple groups.

The solution is split into a front-end, a back-end and a database. This is for easy separation of the mechanics of the system, as well as for security reasons.

The front-end was built in React js, and it is a website which allows the users to interact with the application in a reasonable manner.

The back-end was built using Node js. It was built in an api format using the express framework. The reason for this is for ease of communication with the front-end, as well as the ability to expand to having multiple front-ends in the future.

The database was built using mongodb. This was mostly because of the ease of communication between a node server and mongodb. The liberty to use a nosql database was due to the database not needing to record transactions in a specific order and the relatively small size of the project.

Front End

The front end is a website built in React js. It acts as an interface for the application.

The application is composed of 4 pages:

- The registration page
- The login page
- The wall
- The groups admin page

Registration Page

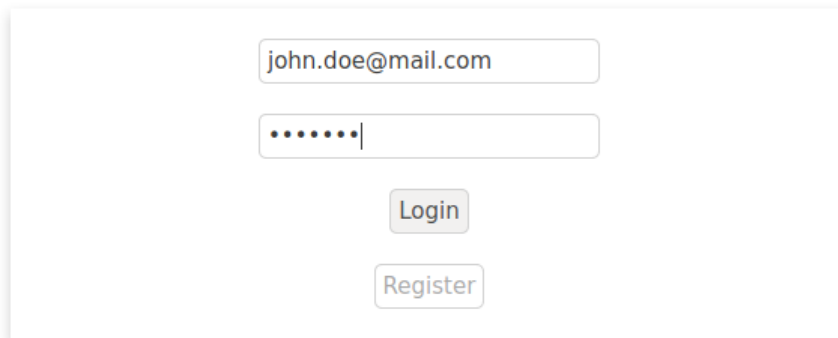
This page allows a user to register for the application. It is composed of a form which takes in a name, email and password. There is some basic validation in the front-end. Then if that validation is passed these pieces of information are sent to the back-end. It will also do some validation. If a requirement is failed, the front-end will get a message and display the reason for the failure, so that the user may correct it. If all of the values are correct, the registration will go through and the registration page will redirect to the login page.

A white registration form with a light gray border and a subtle shadow. It contains three input fields: the first has "John Doe", the second has "john.doe@mail.com", and the third has seven dots. Below the fields are two buttons: "Register" and "Login".

John Doe Registering

Login Page

This page allows an existing user to access the application. It is composed of a form that accepts an email and password. There is some basic validation in the front-end. If the information is valid it is sent to the back-end. There the information has some more validation, if some of the validation is not passed, the message is returned and displayed in the front-end. If the validation is passed, the given email and password are checked against existing users. If one is not found, an error is returned, otherwise a json web token is returned which allows the user to use features that only exist if logged in.

[Wall](#)[Groups](#)[Log Out](#)

A login form with a white background and a subtle shadow. It contains two input fields: the first for an email address, which has 'john.doe@mail.com' entered, and the second for a password, which has seven dots and a cursor. Below the password field are two buttons: 'Login' and 'Register', both with a light gray background and rounded corners.

John Doe Logging In

The Wall

This page has two functionalities:

- making new posts
- seeing previous posts

[Wall](#) [Groups](#) [Log Out](#)

Besties

Post

`bab3d01bf674a22abf8ee5caf4811e3b3fdc9cafffe292a291991e9128e37c6bb2a66edfdeabc78ad619d989349e95ef456de9c53de7e021b8c6ae29d4e20c27eadacbe0d8ed057a750800ca3c069231d7fbab65175744b25658bb6dda4045cb89120fb9cfa0d0e5d`

`994aefb50b5cb34a203d05c0facd9f3f3a07f40f70846683f5e4b9ac1295c35aef4abdd8fa4b1954b488742dd4b9bdd815bf22089b73bc21d89e6433f244c9c7fac5aba6d1d6ce3c73cea7b0724fc59c49497e28220c136c403531616c6bc9b10a7e7fde952f0f42682a5dba6844a2f35756a2b1943485cb5f595d54313b`

2020-04-14T16:23:39.816Z

Best post is my post

This post is for all of my friends out there

2020-04-14T16:23:39.816Z

Jane Doe's Wall - she was added to the *My Friends* group

New Posts

At the top of the page there is a form. This form accepts a title, some content and a group to post for. There is some basic validation. If this validation is passed, the information is sent to the back-end where the post is stored.

Past Posts

Underneath the form, all of the previous posts on the platform are shown. By default the posts are shown as cyphertext, i.e. they are not decrypted, they are simply sent in their encrypted format. The other posts are posts from groups the user is a member of. These are decrypted and shown to the user in plain text.

Groups Admin Page

This page has two functionalities:

- creating new groups
- administration of current groups

A white card with a rounded shadow. It contains a text input field with the value "My Family" and a button labeled "Add Group" below it.A white card with a rounded shadow. It has the title "Wide Boyz" in bold. Below the title is an "Email" input field, an "Add User" button, and a "Leave Group" button.A white card with a rounded shadow. It has the title "My Friends" in bold. Below the title is an input field containing "jane.doe@mail.com", an "Add User" button, and a "Leave Group" button.

John Doe's Groups

New Groups

At the top of the page there is a form which simply accepts a name. This name is sent to the back-end which creates the group and makes that user a member of the group.

Current Groups

Underneath the form, there is a set of cards, one for each group the user is a member of. These cards allow administration of the groups in two ways. They allow the user to remove itself from the group. As well as add other users to the group by email.

Back End

task as understood
technologies
approaches
libraries
improvements