BACHELORARBEIT

FACHBEREICH 4: INTERNATIONALE MEDIENINFORMATIK

Thunderbird: One Time Password

Student
Esteban LICEA
Matr. Nr. 536206

Primary Mentor
Prof. Dr. Debora
WEBER-WULFF

Secondary Mentor
Prof. Dr Kai Uwe
BARTHEL

June 13, 2022

Contents

Chapter 1

Introduction

The digital age has fully taken hold into our societies. We do everything in some for or another of digital media: create art, science, communicate, create and share memories, play games, write thesis reports. There is basically no limit to what people do with their computers.

Related to this, the growth of the internet has more and more pushed our activities online. In it's nascent, this was not thought to be a major as things like Chinese or Russian Hackers, or even U.S. government intrusions were not really thought to be more than fringe blogger material. Especially, and likely most damaging, was the basic expectation of private communication. Edward Snowden's revelations about the "Five Eyes" intelligence alliance, and cooperation in the collection of all online communication, social media, phone data, etc. No online communication has been considered safe ever since.

Mozilla has tried to support end-to-end encryption (E2EE? for a long time, it has been faced with a major obstacles:

- Setting the PGP add-on Enigmail was too technical
- Generating keys was too technical
- Even if conditions 1. & 2. were fulfilled, it was especially uncommon that anyone else you would want to converse with would have gone through the trouble to setup a client or keys for themselves
- Mozilla is in the process of using OpenPGP build-in to the client, but that also has problems, most obviously, you again need new keys (granted easier to setup this time)

- and, again, both people must have generated keys (again

This project is centered on the implementation of an Email Add-on that will allow end-to-end encrypted (E2EE) communication. More specifically, it will focus on the Mozilla Thunderbird client, for the simple fact that I have personally used it for over ten years, it's free, open-source, and cross platform. While I grant that not everyone uses Thunderbird, at least there should be no shortage of users, and theoretically anyone can get it easily, for free.

Ultimately, this project aims to offer a simple, albeit *not* perfect solution for those interested in privacy, that don't have the technical expertise to engage in key creation, exchanges or have zero knowledge about encryption. The will demonstrate the advantages and disadvantages of various implementations strategies, and implement a solution that offers, hopefully, a viable encryption option that will fulfill some use cases.

Chapter 2

Cryptography

2.1 Choosing Algorithm

Chapter 3

Implementation

3.1 WebExtensions

3.2 Crypto JS

Chapter 4

Security Considerations

4.1 Attack Vectors

4.2 Attack Mitigation

Chapter 5

Summary

Chapter 6

Software Requirments Specifications

6.1 Introduction

6.1.1 Purpose

This document will describe the entire software development process, including use cases, personas, diagrams, and the end goals of the system. The audience for this document will be any persons interested in the software engineering process used for this project, but more specifically, those responsible for overseeing and rating this project.

6.1.2 Scope

The name for this product will be "Thunderbird: One Time Password." This product will be a Thunderbird add-on, that will encipher plain text into cipher text, which will be delivered by the Thunderbird client to another Thunderbird recipient, that also has the add-on installed. Finally, the second person will be able to decipher the cipher text back to plain text, and read the message.

6.1.3 Definitions, acronyms, abbreviations

The following definitions, acronyms, and abbreviations may be used with in the software development process:

client Refers to an email client, more specifically Mozilla's Thunderbird email client.

E2EE End-to-end encrypted, in this case, an end-to-end encrypted email.

JS JavaScript.

AES Advanced Encryption Standard.

IEEE Institute of Electrical and Electronics Engineers.

asymmetric encryption Encryption that only uses one key for encryption.

symmetric encryption Encryption that requires two keys, one on each side of the private message exchange.

API application programming interface.

extensions An extension adds features and functions to a browser.

plain text The text that we wish to encrypt.

cipher text The encrypted text.

ECB Electronic Codebook, a AES encryption mode.

CBC Cipher Block Chaining, a AES encryption mode.

CFB Cipher Feedback Mode, a AES encryption mode.

OFB Output Feedback Mode, a AES encryption mode.

CTR Counter Mode, a AES encryption mode.

SRS Software Requirements Specification.

6.1.4 References

Author used the IEEE document:

1. IEEE Std 803-1998

the IEEE Recommended Practice for Software Requirements Specifications. [1]

[1] https://cse.msu.edu/ cse870/IEEEXplore-SRS-template.pdf

6.1.5 Overview

6.2 Overall Description

The following subsections will describe the general factors that will influence the product requirements, including any background information.

6.2.1 Product perspective

The developed software product, *Thunderbird: One Time Password*, has not current rival. It current alternatives would be Mozilla's own implementation of OpenPGP. The previous option was PgP through the add-on Enigmail. However, at the writing of this document, the add-on is no longer supported.

The two alternatives do have the advantage that they used symmetric key exchange to encrypt emails, which is more secure, and recommended for encoded email exchange. The *Thunderbird: One Time Password* add-on will have the feature that it is easy to use, at the expense of security.

System interfaces

The required, and assumed interfaces required for the product include the following:

1. A modern system, running one of three operating systems:
   (a) Windows 10 or later
   (b) Apple running Big Sur or later
   (c) Linux variant, running a modern system
2. an Internet connection

User interfaces

There are no special user interface requirements.

Hardware interfaces

There are no special hardware interfaces required for this product to function.

Software interfaces

The required software interfaces are:

1. Mozilla's free, open source email client, Thunderbird, to be installed on the system.

2. The client should be configured to send and receive emails.[2]

3. The client should be updated to the latest current software version.

4. The client can be installed on any current (or recent) Windows, Linux, or Apple OS.[3]

Communications interfaces

No special communication interfaces will be required, than would already be prerequisites for Email communication, i.e. network capable computer.

Memory constraints

Not applicable

Operations

Not applicable

Site adaptation requirements

Not applicable

[2]Thus, an email account on an email server is assumed.
[3]No other OS will be tested.

6.2.2 Product functions

6.2.3 User characteristics

6.2.4 Constraints

There will be various constraints within this project listed below:

- Security: It will not be possible to account for all attack vectors. Thus, only known, common attack vectors will be discussed.
- Security: How Mallory comes into possession of an encrypted email may not be fully explored. Related to 1. above, but we'll at least give an examination to this possibility – however she came into possess the Email.

6.2.5 Assumptions and dependencies

6.3 Specific Requirements

6.3.1 Use Cases

The Use Cases used in this project will be defined, and or be restricted to the following items:

Use Case ID

The Use Case ID will be a unique, numeric identifier for the use case.

Actor(s)

An actor is a person or other entity external to the system who interacts with it, and performs use cases to complete task. Included in this designation, will be additional actors who participate in the use case.

Description

This section should describe at a high level the purpose of the use case, what it aims to achieve, and any other relevant outcomes.

Preconditions

The preconditions are all those conditions that must exist prior to the execution of the use case.

Basic Flow

These are the basic, ordered steps and the description required for the completion of the use case. The steps will be numbers, and should be executed in this exact order. Completing the steps, in this order, should lead to the completion of the use case without error.

Exceptions

Describes any anticipated errors that could occur during the execution of the use case, and how the system will handle these errors. The exceptions systems will not describe unanticipated errors, or error that are not included in the basic flow.

Postconditions

Describes the state of all relevant parties, including the system, *after* the execution of the use case.

Use Case ID:	0
Actor(s):	Alice
Description:	Alice will encrypt an email to Bob
Preconditions: 1. Thunderbird Email client installed. 2. Thunderbird Email client configured to send and receive emails. 3. Super-duper Addon installed. 4. Email written	
Basic Flow: 1. Alice writes an email in Thunderbird. 2. Alice locates and click on the add-on button. 3. Observe: Alive sees a popup screen encrypt the email. 4. Alice is prompted to enter a password.	
Exceptions: 1. N/A	
Postconditions: 1. The email is enciphered. 2. The addon window closes. 3. Alice is returned to the Thunderbird client.	

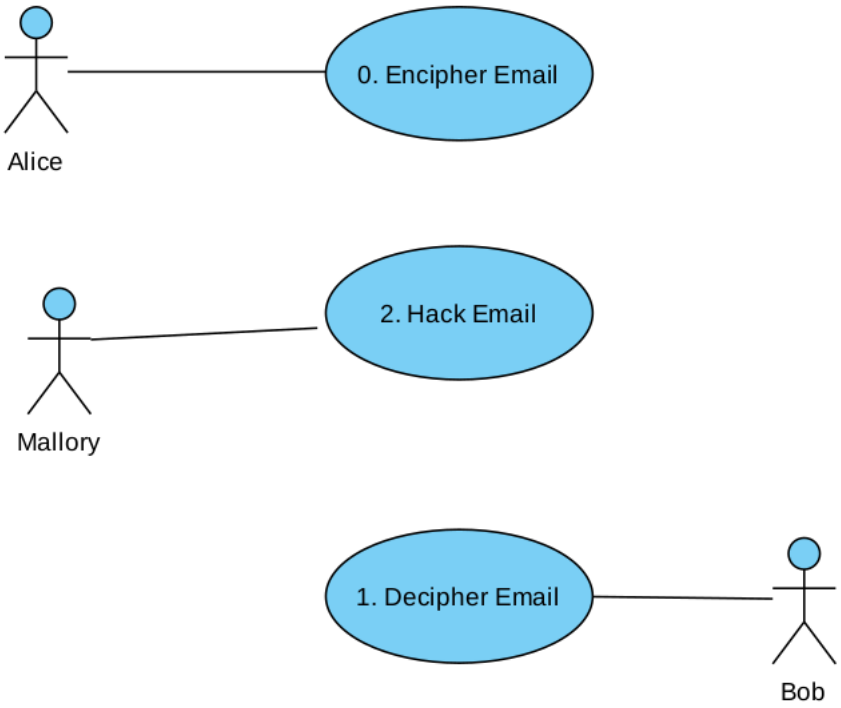
Use Case ID:	1
Actor(s):	Bob (or any other intended recipient) decrypts an Email from Alice
Description:	Alice will encrypt an email to another actor, then share a password with them, that they will then be able to decrypt
Preconditions: 1. Thunderbird Email client installed. 2. Thunderbird Email client configured to send and receive emails. 3. Super-duper Add-on installed. 4. Email written	
Basic Flow: 1. Alice writes an email in Thunderbird. 2. Alice locates and click on the addon button. 3. Observe: Alive sees a popup screen encrypt the email. 4. Alice is prompted to enter a password. 5. Alice enters a password. 4. Alice shares this password with said actor *offline*.	
Exceptions:	

1. N/A
Postconditions: 1. The email is enciphered. 2. The addon window closes. 3. Alice is returned to the Thunderbird client.

Use Case ID:	2
Actor(s):	Mallory
Description:	Mallory will decipher an email
Preconditions: 1. Thunderbird Email client installed. 2. Thunderbird Email client configured to send and receive emails. 3. Super-duper Addon installed. 4. Email written	
Basic Flow: 1. Alice writes an email in Thunderbird. 2. Alice locates and click on the addon button. 3. Observe: Alive sees a popup screen encrypt the email. 4. Alice is prompted to enter a password. 5. TBD.	
Exceptions: 1. None allowed. =)	
Postconditions: 1. The email is still enciphered. 2. The add-on window closes. 3. Mallory is returned to the Thunderbird client.	

6.3.2 Use Case Diagrams

Use Case Diagrams



6.4 Appendix