Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

BACHELORARBEIT

FACHBEREICH 4: INTERNATIONALE MEDIENINFORMATIK

Thunderbird: One Time Password

Primary Mentor
Prof. Dr. Debora
WEBER-WULFF

Student
Esteban LICEA
Matr. Nr. 536206

Secondary Mentor
Prof. Dr Kai Uwe
BARTHEL

June 13, 2022

Contents

Chapter 1

Introduction

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

Proportional to this growth, the internet's influence on our lives has also ballooned. Our activities have been pushed more and more online, onto the cloud. Originally, few bothered to think about privacy. Most damaging, perhaps, was the erroneous 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, and phone data. No online communication has been considered safe ever since.

1.1 Problem

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

Thus, the problem: How can Alice send an encrypted email to someone that does not have any type of public key available?

1.2 Context

While PGP has existed for years, it is predicated on the exchange of public keys. In clear text, there is a technical requirement to create and exchange keys, and installation of any additional required client software that most average users do not have the patience to complete. Originally, Thunderbird relied on an add-on, Enigmail, to create, manage, and exchange keys.

Starting with Thunderbird 78, Mozilla implemented OpenPGP as part of it's core client software, and dropped support for all add-ons not using MailExtensions (which includes Enigmail). However, the feature is disabled by default, and is still considered a work in progress. All other add-ons found on Thunderbird's extensions page or searching through Github were considered to be in a testing or experimental phase.

1.3 My solution to the problem

This project will implement 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.

1.4 Methods applied

The methods and tools used to solve this research inquiry will include:

1. Literature either in the form of online or paper publications, i.e. books
2. Online learning resources
3. Thunderbird and JS Encryption APIs
4. Guidance from Mentors
5. Visual Studio Code for code production
6. Github for Source Code and Thesis code management
7. Latex for writing the Thesis
8. Jira for project management, i.e. Kanban board, sprints, and road maps

After the research has been completed, all coding will proceed using a test driven development approach. Thunderbird Add-ons are based on MailExtension technology, which are created using the follow standard languages:

1. HTML
2. CSS
3. Javascript

Chapter 2

Cryptography

2.1 Algorithm selection overview

2.1.1 Symmetric key encryption

Selecting an algorithm, among so many, was pretty straightforward given my use case, but I wanted to show my thought processes. Firstly, there are two fundamental paths for selecting an encryption algorithm. The selection between *asymmetric* and *symmetric* key encryption is the initial decision.

1. Asymmetric-key cryptography: A public and private key are created by both people wanting to exchange encrypted emails. This is the most secure and most commonly implementated encryption available, popularly known as "public-key encryption." Examples encryption key algorithms used include RSA and Diffe-Hellman-Merkle. There are challenges though:[**?**]

   (a) Both parties need to create their own keys

   (b) Keys need to be exchanged, i.e. a person has to be acute enough to search for the other person's public key – assuming one even exists

   (c) Additional client software is also often required

2. Symmetric-key Encryption: use the same key for both encryption and decryption[**?**, p. 155]

   (a) The primary drawback is that both parties will need to exchange that key, often times in the form of a password.

The goal of this project is *ease of use* (at the cost of security), so our choice is clear: symmetric-key encryption.

2.1.2 Block vs. Stream cipher encryption

Next, we need to decide between a block cipher or a stream cipher. As Bruce Schneier defines the two in his book "Applied Cryptography: Protocols, Algorithms in C" as:

> There are two basic types of symmetric algorithms: block ciphers and stream ciphers. Block ciphers operate on blocks of plaintext and ciphertext—usually of 64 bits but sometimes longer. Stream ciphers operate on streams of plaintext and ciphertext one bit or byte (sometimes even one 32-bit word) at a time. With a block cipher, the same plaintext block will always encrypt to the same ciphertext block, using the same key. With a stream cipher, the same plaintext bit or byte will encrypt to a different bit or byte every time it is encrypted.[**?**, p. 12]

The advantages of a stream ciphers: [1]

- bit or byte at a time encryption
- speed of encryption/decryption

are more appropriate for hardware implementations.

According to Bruce Schneier, block ciphers are more suitable for software implementation as they are easier to implement, avoid time-consuming bit manipulations, and operate on computer sized blocks. [**?**, p. 172]

2.1.3 Block cipher selection

There are many block ciphers to choose from, these are just some of the most popular: [**?**]

1. Digital Encryption Standard(DES): DES is a symmetric key block cipher that uses 64-bit blocks, but it has been found vulnerable to powerful attacks. This is the reason the use of DES is on a decline.

[1] https://crashtest-security.com/block-cipher-vs-stream-cipher/

2. Triple DES:This symmetric key cipher uses three keys to perform encryption-decryption-encryption. It is more secure than the original DES cipher but as compared to other modern algorithms, triple DES is quite slow and inefficient.

3. Advanced Encryption Standard(AES): AES has superseded the DES algorithm and has been adopted by the U.S. government. It is a symmetric key cipher and uses blocks in multiple 32 bits with minimum length fixed at 128 bits and maximum at 256 bits. The algorithm used for AES, was originally named Rijndael.[2]

4. Blowfish: Blowfish is a symmetric key block cipher with a block size of 64 and a key length varying from 32 bits to 448 bits. It is unpatented, and the algorithm is available in the public domain.

5. Twofish: Twofish is also a symmetric key block cipher with a block size of 128 bits and key sizes up to 256 bits. It is slower than AES for 128 bits but faster for 256 bits. It is also unpatented and the algorithm is freely available in the public domain.

But, after an overall account of the block enciphers available, the author decided there is really only one reasonable choice: the Advanced Encryption Standard (AES). This decision is based

2.2 Advanced Encryption Standard (AES)

[2]The winners of the AES competition were two Belgians: Vincent Rijmen, Joan Daemen, thus the algorithm's name: "**Ri**n**jdae**l"

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

6.4 Appendix