DISTRIBUTED FILE SYSTEM BY ADVANTAGING ONLINE WEB SERVICES

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

by

Glenn Olsson

June 2022

COMMITTEE MEMBERSHIP

TITLE: Distributed file system by advantaging online web services

AUTHOR: Glenn Olsson

DATE SUBMITTED: June 2022

COMMITTEE CHAIR: Zachary Peterson, Ph.D.
Professor of Computer Science

COMMITTEE MEMBER: Aaron Keen, Ph.D.
Professor of Computer Science

ABSTRACT

Distributed file system by advantaging online web services

Glenn Olsson

Today there are free online services that can be used to store files of arbitrary types and sizes, such as Google Drive. These services are often limited by a certain total storage size. My goal is to create a filesystem that similarly can store arbitrary amount and types of data but without any real limit. This is to be achieved by taking advantage of online webpages such as Twitter where text and files can be posted on free accounts with no visible limit. The goal is to have a filesystem that behaves like any other but where the actual data is stored for free on unsuspecting websites.

ACKNOWLEDGMENTS

Thanks to:

- Andrew Guenther, for uploading this template
- I would like to thank xxxx for having yyyy.

TABLE OF CONTENTS

APPENDICES

LIST OF TABLES

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

1.1 Project Overview

This project intends to create a filesystem by taking advantage of online web services such as twitter. The idea is to save the files by posting/sending an encrypted version as one or more posts/private messages on for instance Twitter. The goal is to achieve storage of data in the same scale as the free accounts on online storage services such as Google Drive where users can store up to 15Gb of files. Accomplishing this would mean that one can store more data than that for free using this new filesystem.

The intention is not to create a revolutionary fast and usable filesystem but to instead to explore how well it is possible to utilizing the storage that Twitter and similar services provides by allowing users to post text and files, almost unmonitored.

The data posted will be encrypted and not be comprehensible by anyone who would stumble upon a post. Preferably, you should not be able tell that the post is used for storage of data at all.

Chapter 2

BACKGROUND

This chapter provides basic background information about xxx. Additionally, this chapter describes xxx. The chapter also describes related work xxxx.

2.1 Filesystems

Filesystems are used to store data on for instance a hard drive of a computer. Google Drive is another file system that enables user to save their data online up to 15 GB for free[1] using their clusters of distributed storage devices, meaning that the data is saved on theirs servers which can be located wherever[2]. Paying customers can achieve higher amount of storage using the service.

A deniable filesystem is a system that does not expose files stored on this system without credentials - neither how many files are stored, their sizes, their content or even if there exists any files on the filesystem[3]. This is useful if for example one is to be exposed to an audit of their data by a totalitarian regim where they don't even want to disclose that they have data.

2.2 Twitter

Twitter is a micro-blog online where users can sign up for a free account and create public posts using text, images and videos. Text posts are limited to 280 characters while images can be up to 5mb and videos up to 512mb[4]. There is also possibility to send private messages to other accounts, where each message can contain up to 10'000 characters and the same limitations on files. If one would represent an arbitrary file of X bytes, each byte (0x00 - 0xFF) can be represented as a character and we can therefore represent this file as X different characters. Using the same set of characters for encoding and decoding we can get a symmetric relation for representing a file as a string of characters. This text can theoretically be posted on for instance Twitter, as long as the size is smaller than 280 or 10'000 bytes depending on if we would post a public post or a private message.

2.3 Related Work

Peters created a deniable filesystem using a log-based structure in 2014[3]. The filesystem of my project could be seen as a deniable system in the sense that the data is not actually stored on the device, and if the filesystem is not mounted it could be hard to prove that the user actually has data, even if they for instance would find the twitter account. This was also developed using FUSE[5] which I also will be using.

Chapter 3

METHOD OR METHODS

Chapter 4

WHAT YOU DID

Chapter 5

RESULTS AND ANALYSIS

Chapter 6

DISCUSSION

Chapter 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

7.2 Limitations

7.3 Future work

Due to the breadth of the problem, only some of the initial goals have been met. In these section we will focus on some of the remaining issues that should be addressed in future work. ...

7.3.1 What has been left undone?

The prototype does not address the third requirment, i.e., a yearly unavailability of less than 3 minutes, this remains an open problem. ...

7.3.1.1 Cost analysis

The current prototype works, but the performance from a cost perspective makes this an impractical solution. Future work must reduce the cost of this solution, to do so a cost analysis needs to first be done. ...

7.3.1.2 Security

A future research effort is needed to address the security holes that results from using a self-signed certificate. Page filling text mass. Page filling text mass. ...

7.3.2 Next obvious things to be done

In particular, the author of this thesis wishes to point out xxxxxx remains as a problem to be solved. Solving this problem is the next thing that should be done. ...

7.4 Reflections

One of the most important results is the reduction in the amount of energy required to process each packet while at the same time reducing the time required to process each packet.

The thesis contributes to the numbers 1 and 9 by xxxx.

Bibliography

[1] *Cloud Storage for Work and Home – Google Drive*. sv. URL: https://www.google.com/intl/sv/drive/ (visited on 10/26/2021).

[2] *Distributed Storage: What's Inside Amazon S3?* en-US. URL: https://cloudian.com/guides/data-backup/distributed-storage/ (visited on 10/26/2021).

[3] Timothy Peters. "DEFY: A Deniable File System for Flash Memory". In: *Master's Theses*. June 2014. DOI: 10.15368/theses.2014.76. URL: https://digitalcommons.calpoly.edu/theses/1230.

[4] *Media Best Practices - Twitter*. en. URL: https://developer.twitter.com/en/docs/twitter-api/v1/media/upload-media/uploading-media/media-best-practices (visited on 10/26/2021).

[5] *libfuse*. original-date: 2015-12-19T20:27:34Z. Oct. 2021. URL: https://github.com/libfuse/libfuse (visited on 10/26/2021).

APPENDICES

Appendix A

SOMETHING EXTRA

A.1 Just for testing KTH colors

This will be an appendix