FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Review report of a final thesis

Reviewer:	Ing. Josef Kokeš, Ph.D.
Student:	David Košťál
Thesis title:	Peer-to-peer Backup Application
Branch / specialization:	Computer Security and Information technology
Created on:	29 May 2023

Evaluation criteria

1. Fulfillment of the assignment

▸ [1] **assignment fulfilled**
[2] assignment fulfilled with minor objections
[3] assignment fulfilled with major objections
[4] assignment not fulfilled

The assignment is way too complicated for a bachelor thesis. Essentially, the work consists of no less than 6 major subsystems: generating and restoring backups, peer discovery, efficient communication with the peers, fail-safe storage, authentication, data encryption. All of these, plus a host of lesser components, need to be not only researched, but also implemented. The student touched on all of these topics, but due to the sheer scope and complexity of the tasks, coupled with the limited amount of resources available for a thesis, it's inevitable that he couldn't solve them thoroughly. This makes grading very difficult and rather inconsistent - one thing is, what is necessary to create a practically usable application, quite another, what is reasonable to expect in a thesis project. Generally, I decided to grade based on the amount and quality of the work rather than its completeness, but readers are warned - don't assume the high marks mean the application is safe to use!!

2. Main written part 90/100 (A)

As noted in the previous point, the scope is too big! As a result, the text of the thesis, while fairly detailed and quite well thought out, barely touches on the aspects that need to be handled in detail in a work of this type - and with each of them, major issues remain unsolved. There are too many to name, so just a sample from the actual backup part (disregarding any other aspects): How does the application even decide which data needs to be backed up? How is file access handled? It seems there's an assumption that file data becomes available in zero time, but what if the data changes while it is being read? There is no provision for possible stalled backup processes and no resolution for a new backup task triggering before the last one had a chance to complete. Special file types (links, offline files, sparse files...) are never discussed. The restore process is vague at best. The user is completely ignored in all of this, beyond claims that users consider backups difficult and that's why they don't use them. Etc.

Since the student's branch is security, a few warnings there: The threat model, even though it takes up almost 12 pages, is by far not complete. For example, the problem of malicious actors (all of them - the user, the peers, the server) is generally ignored and the actors are considered if not trustworthy, then at least inclined to cooperate; the possibility of malicious actors is, for the most part, limited to users trying to send potentially dangerous/offending/illegal content to a victim. I am very dubious about the use of a simple PRNG for the generation of secrets (figure 5.1), a good KDF (key derivation function) seems more appropriate. The fact that authentication is done through a custom protocol is a major reason for worry - why wasn't a standard mechanism (e.g. client and server certificates) used instead? Also, there doesn't seem to be any provision for the inevitable future evolution, e.g. of the encryption or authentication schemes. Why does the server need to know a nonce - that could have been encrypted so that only the actual peer would be able to decrypt it. Etc.

That said, I want to stress that the thesis is not at all bad - in fact, it is far better than I would expect, with many important topics at least considered, if not resolved. It's just that the limited space and time available for a thesis prevents the student from providing a complete reliable solution. I would have much preferred if the student focused on only one aspect but performed a really detailed and reliable analysis of it, including leaving out all implementation beyond a simple proof-of-concept. That should include a far more detailed review of known techniques, in any peer-to-peer data sharing schemes I expect at least a mention of Bittorrent or Freenet or other mature implementations.

The text is generally well written and follows a logical sequence which makes it easy to understand. The figures in it help a lot in conveying this understanding. Some minor issues remain (e.g. no linking text between sections 5.4 and 5.4.1, frequent switching of verb tenses between past, future and present, some missing articles), but that's not a real problem.

3. Non-written part, attachments 85 /100 (B)

The project is rather extensive, containing both a server and a client part, both of which are split into a Javascript-based frontend and a Rust-based backend. Due to the sheer scope, I focused only on certain parts, particularly those dealing with security.

The code is generally well written and documented and follows recommended coding practices (e.g. established libraries rather than a custom code are used for most of the functionality, database is accessed using prepared statements). The incorrect code mostly stems from the deficiencies in the analysis, see above. A few places that I consider suspect beyond that:

- It seems the root secret is stored in plaintext in the client's configuration database, which is definitely not a recommended practice.
- The keys should be derived from the root secret using a key derivation function rather than a seeded PRNG.
- The obfuscation key gets newly generated not only during the initial setup, but also when restoring a previous setup, which invalidates stored data packets belonging to other users, making their backups unrecoverable!!
- Pack headers and other metadata apparently all use the same encryption key. A recommended practice is to use a random salt for each object.

4. Evaluation of results, publication outputs and awards 49 /100 (F)

While I am convinced that the concept is sound and that the application will eventually become what it tries to be, I don't consider the current results to be actually usable. This is chiefly due to the problems outlined in the evaluation of the main written part. A backup application must be first and foremost reliable and unfortunately the unresolved issues prevent that, especially in the long term. What we have now is a nice demo of the future possibilities, but .I warn against using it in practice.

The overall evaluation 88 /100 (B)

I have been struggling with the evaluation of the thesis. I have no complaints about the quality of the student's work or the amount of it, both in the text and in the code - all of that is really good. On the other hand, what I consider a major feature of any backup tool is the reliability and the thesis rarely even touches that subject, much less solves it satisfactorily. That casts a strong doubt on the stated goal of creating foundations for a P2P backup application - I am afraid the current foundations are too shaky and will need to be torn down and rebuilt almost from scratch. In the end, I decided to grade the thesis based on what it contains rather than on what the finished product needs, but I warn the reader not to take the high marks as a recommendation to use the results of the work in practice.

Questions for the defense

How much of the work is new and how much is derived from the previous version of your application?
Which aspects do you consider the most critical in the current state of the thesis?

Instructions

Fulfillment of the assignment

Assess whether the submitted FT defines the objectives sufficiently and in line with the assignment; whether the objectives are formulated correctly and fulfilled sufficiently. In the comment, specify the points of the assignment that have not been met, assess the severity, impact, and, if appropriate, also the cause of the deficiencies. If the assignment differs substantially from the standards for the FT or if the student has developed the FT beyond the assignment, describe the way it got reflected on the quality of the assignment's fulfilment and the way it affected your final evaluation.

Main written part

Evaluate whether the extent of the FT is adequate to its content and scope: are all the parts of the FT contentful and necessary? Next, consider whether the submitted FT is actually correct – are there factual errors or inaccuracies?

Evaluate the logical structure of the FT, the thematic flow between chapters and whether the text is comprehensible to the reader. Assess whether the formal notations in the FT are used correctly. Assess the typographic and language aspects of the FT, follow the Dean's Directive No. 52/2021, Art. 3.

Evaluate whether the relevant sources are properly used, quoted and cited. Verify that all quotes are properly distinguished from the results achieved in the FT, thus, that the citation ethics has not been violated and that the citations are complete and in accordance with citation practices and standards. Finally, evaluate whether the software and other copyrighted works have been used in accordance with their license terms.

Non-written part, attachments

Depending on the nature of the FT, comment on the non-written part of the thesis. For example: SW work – the overall quality of the program. Is the technology used (from the development to deployment) suitable and adequate? HW – functional sample. Evaluate the technology and tools used. Research and experimental work – repeatability of the experiment.

Evaluation of results, publication outputs and awards

Depending on the nature of the thesis, estimate whether the thesis results could be deployed in practice; alternatively, evaluate whether the results of the FT extend the already published/known results or whether they bring in completely new findings.

The overall evaluation

Summarize which of the aspects of the FT affected your grading process the most. The overall grade does not need to be an arithmetic mean (or other value) calculated from the evaluation in the previous criteria. Generally, a well-fulfilled assignment is assessed by grade A.