CS CAPSTONE WINTER MIDTERM PROGRESS REPORT

FEBRUARY 16, 2018

PRIVACY PRESERVING CLOUD, EMAIL, AND PASSWORD SYSTEMS

PREPARED FOR

OSU

ATTILA YAVUZ _____ _____
Signature *Date*

PREPARED BY

GROUP 38

THE SECRET BUNNY TEAM

ANDREW EKSTEDT _____ _____
Signature *Date*

SCOTT MERRILL _____ _____
Signature *Date*

SCOTT RUSSELL _____ _____
Signature *Date*

Abstract

This document provides a report of progress our team has made on our project during the first half of the Winter 2018 term. We outline our current status, problems we've encountered, and future work. We also provide individual accounts of what each group member has accomplished.

CONTENTS

1 Project Purpose and Goals

The "Privacy Preserving Cloud, Email and Password Manager" Capstone project is a research-oriented project that aims to find a way to implement the DSSE scheme proposed by David Cash. This implementation will be executed through command line prompts and hosted on OSUs engineering servers. A user can use this system with a client-server model to perform actions, such as search or update, on a "cloud-based" database. User interface is not considered a priority as this project is not intended to be used in any commercial capacity. There are five primary goals of this project. First the implementation of David Cash's DSSE Algorithm. The following four aspects can be done in parallel: parallelization, optimization, cloud integration, and email integration. Finally benchmarking will be done during the final stages and after implementation of the previous items.

2 Current Status

Our original schedule called for us to work on the core DSSE scheme for the first two-to-three weeks of Winter term, and then branch off into projects using that core. So that's what we worked on. At the advice of our client, we took a couple more weeks to add more features to the core and fill it out a bit before moving on to other stuff.

We have the following methods implemented

- Setup - create an initial encrypted search index from a set of documents
- Search - search for a token in the encrypted database
- Add - update the encrypted index by adding tokens to a file

Also:

- network connectivity over local socket

3 Problems that impeded progress, with solutions

As we begin implementation of our capstone project it is important to be able to look back at progress throughout the term. If we can understand what went well and poorly we can use that information to be able to have a more cohesive plan for the following terms.

3.1 Establishing a meeting time with our client

The first time our client, Attila, was available to meet with us was in Week 3, and it turned out that he got sick that week so we were unable to meet until Week 5.

Going forward, we are planning to meet every approximately every two weeks.

3.2 Missing TA

Our capstone TA was missing in action for the first few weeks of the term. Combined with the difficulty of meeting with our client, this left us with little outside guidance for the first half of the term.

We were eventually able to work out a meeting time in Week 5.

3.3 Group Dynamic

In the fall term we had a clear cut goal for each week. With a new writing assignment assigned and due it kept us on a good pace with what we imagined the project process would be. Now that we are ”cut lose” to do our own implementation we are finding it hard to stay on track compared to our Gantt Chart that we imagined progress would be this term.

4 Future work

We have a lot of work ahead of us. We have a basic DSSE implementation in place now that we can launch from, but there is still a lot more left to implement.

- Implement two-level pointer optimization described in [1]
- Write email integration daemon
- Write persistent storage module for DSSE
- Investigate storage-only server
- Investigate parallelism
- Benchmark our implementation versus IM-DSSE and Clusion
- Write up our results

5 Individual Progress

5.1 Andrew Ekstedt

5.1.1 Overview

The first half of the term for me was mostly about implementing the core DSSE methods, and the client and server communication. I was able to get three of the four methods implemented: Setup, Search, and Add.

Last term, I wrote a little Python prototype of the DSSE to help understand the protocol, which was super helpful. When implementing the C++ version, I was able to, to some extent, transliterate the Python code line-by-line, allowing me to focus more on the low-level details of implementation rather than also having to hold the high-level details of the algorithm in my head.

We had a little trouble getting in touch with our TA and client initially, but we now have set up weekly meetings with out TA and biweekly meetings with our client.

Things have not gone as quickly as we had hoped, but our client seems understanding, and I think we’re in a good position to at least get the core of what we are trying to accomplish done. Going forward, while there is still a little more work to do on the core DSSE, we should be able to start some of the optimizations and other projects that we have planned.

5.1.2 Week 1

Progress:

Start of the term! This week was mostly about getting back into the swing of school after Winter Break, getting back up to speed with what we had done last term, and planning meetings with our group, our client, and our TA.

We scheduled some group meeting times in week 2 (and have continued to meet at the same time throughout the rest of the term). We reached out to our client to schedule a meeting, but the first time he could meet was week 3.

Problems:

- Capstone is at 8 in the morning, but fortunately it doesn't meet often

5.1.3 Week 2

Progress:

The task this week was to start working on the core DSSE implementation. I decided to dive in and stub out the code, and got a simple program compiled and running. The DSSE is factored into Core, Client, and Server classes. The core class is responsible for all the cryptography, and the Client and Server classes are responsible for the network layer communication. I wrote out some header files for each of these classes, and implemented the Setup and Search methods of the core DSSE.

I made a decision early on to vendor all our third-party dependencies into our source repository. The advantage of doing this is that the program can be built out-of-the-box with a single command, without the user having to mess around with compiling and installing a bunch of libraries. This was borne out of direct experience last term with trying to get IM-DSSE to compile. (IM-DSSE is an implementation of another DSSE algorithm that our client had written last year.)

I also helped Scott Russell debug some problems with the mailio library.

Problems:

- Bootstrapping/designing a library from scratch is a lot of work
- It's hard finding time to work on stuff between classes
- Haven't heard from our TA yet

5.1.4 Week 3

Progress:

We met with our client for the first time this week, sort of. Attila had to cancel at the last minute because he was sick, so we met with his grad student Thang Hoang.

I continued to work on core DSSE code. I implemented most of the core Add method, and started implementing some of the networking code for the Client and Server classes. In the DSSE paper [1], the Add and Delete operations are merged into a single Update operation for some reason, even though they are mostly unrelated. I decided to separate them in our implementation.

Problems:

- Client was out sick, so we met with his grad student
- Still haven't heard from TA
- Had some problems getting libraries to build with our project

Summary:

Made some solid progress on the implementation of the core DSSE and the client/server components.

5.1.5 Week 4

Plans:

The plan this week was to get DSSE into a good enough state that we can start building the other stuff. Our original schedule called for us to concentrate on the DSSE implementation for the first two or three weeks of Winter term, after which we would split into working on separate projects.

After talking with Thang last week, it sounds like we want to implement Add as well as Setup and Search.

Progress:

Vendored the ZeroMQ socket library, which allowed me to rip out all my ad-hoc socket-handling code from week 3 and replace it with much more solid ZeroMQ-based sockets.

I implemented the Add method for the core DSSE and the client / server classes.

I also worked with Scott Russell to figure out configure Gmail to deliver the same messages repeatedly over POP3, for testing purposes.

Summary:

Pretty productive week for me: got client & server communication working with zero MQ and implemented the Add method. DSSE is in a pretty good state, just needs persistence.

5.1.6 Week 5

Met with our client, Attila, this week for the first time this term. He seemed satisfied with our progress so far. It seems he is most interested in getting the DSSE written and all the optimizations from the paper implemented, so that we can get some useful benchmark data.

We also met with our TA, Andrew Emmott, this week for the first time. He said it sounds like we are making good progress on our project.

Progress:

I didn't get a whole lot done code-wise. This was due to a combination of having a bunch of work to do for other classes, and having to work on the midterm report for this class.

We worked on our project poster draft, midterm report, and presentation.

Problems:

- Commitments to other classes made it hard to find time for capstone this week. Thing should hopefully calm down soon.

5.1.7 Future work

- Improvements to client and server. The client currently executes some hard-coded operations every time it is run. It needs a command-line interface that allows the user to perform arbitrary actions.
- Implement Delete method. This is the most complicated method of the DSSE scheme, and it touches how the other methods work as well.
- Persistence. The client and server do not currently store any data to disk. This is great for testing, since restarting the server wipes out all the old state, but not so great for actually using actual use. We need to add a storage layer ASAP.
- Storage-only server research. Part of our project is to research whether it is possible to perform all encryption and other computation client-side. This will require working out how to store the underlying data structures such that the client can efficiently retrieve and modify only the parts that it needs.

5.2 Scott Merrill

So far I feel that this term has gone well. The start may have been a bit slow as we had some issues with establishing meeting times, both with our client and our TA. The early morning class meetings were going to be quite rough for me as I have to commute from Eugene. Fortunately we are not going to be meeting every week and therefore my schedule is not as miserable. At the beginning of the term I found my self in a odd situation where I was not sure with what direction I needed to go. The portion I was working on was dependent upon the core DSSE scheme being somewhat functional which took a couple week for that to get accomplished. Now that the core is working, it should be much easier for us to continue to modify it.

5.2.1 Week 1

Plans:

- Review DSSE Scheme for Implementation
- Plan weekly meetings
- First meeting on Sunday

Progress:

- Scheduled a meeting with Attila on January 22nd

Problems:

- Long commute for in class meetings at 8am

5.2.2 Week 2

Progress: Reviewing the David Cash DSSE paper to get insight into goes into the sections I will be working on

Problems:

- Most of the DSSE Scheme pseudocode was written by Andrew in python
- We have not gotten a reply from our TA, will send out another email

Summary: Ready to begin working on implementation, may need to follow along with what Andrew is doing.

5.2.3 Week 3

We met with our clients TA and discuessed progress on implementation as well as some feature we may want to look at (like single sign-on) and needing to add Upadate to the SSE scheme.

Progress: Andrew has made quite a bit of progress with the core algorithm. I have read through the code to understand how I can modularize my components to fit with his.

Problems:

- Have been unable to really write any code yet
- My C++ is quite rusty
- Another email has been sent out to our TA without a response

5.2.4 Week 4

Plans:

- Add tokenization to the DSSE Scheme
- Read up on lvl2 pointers for implementation

Progress:

- Found some resources for how tokenization could be implemented in C++
- Was able to create a basic module for file tokenization.

Summary:

- This week was the first time I was able to push any code to github. I ran some tests on the tokenization method and found that it works as intended. I am not sure when the encryption is called on it, may need to talk with Andrew about how this can fit in with the core scheme he has been working on.

5.2.5 Week 5

Plans:

- Meet with Attila
- Work on DSSE sheme

Progress:

- We met with Attila and learned that we may want to focus on update, delete and disc persistence for the DSSE scheme
- lvl2 pointers optimization was the next task that I plan to take on

Summary:

- This week we met with Attila and were able to have a little better understanding on what is vision for the project was. This was a helpful meeting and it was nice to learn that he seems happy with where we are. We have some new goals to work towards which may have us change priorities on what we focus on first.

5.2.6 Future work

With the main focus of the project being on getting the DSSE scheme running smoothly we have our work cut our for us. Implementation lvl2 pointers, which allow of optimization of the database based on size, will be the next big focus for myself. We also need to have persistence to disc working so that we can store our database rather that having it just run in memory. Also the Dynamic part of this database (having update and delete) have yet to be implemented. I think that all of these things are tasks that can be done simultaneously and we will need to meet to divvy these up. I expect that the next few weeks with be quite intense for our group as all of these parts start to come together in a cohesive whole.

5.3 Scott Russell - Individual Section:

5.3.1 Current Status

This term has been focused on the implementation of our DSSE Algorithm. During the first week we initiated emails to our Client and TA to touch base on progress for the term. It turned out that our Client was sick on the day we planned to meet, and our TA did not respond until week 5. This delayed our planning for the term. While we were waiting to talk to our client in person we worked on the implementation of David Cashs DSSE Scheme.

Personally, my focus so far this term has been on the Email side of the project. First with the research and implementation of an open source C++ POP3 library. In our midterm report presentation, I demoed the basic functionality of this server and explain the setup process creating a test email hosted by Gmail. Email is an important functionality to this project since it allows for this conceptual idea of a DSSE Scheme to be put into a real-world application. I went through 5 different libraries until I ended up with the one we are currently using. (Mailio) It was chosen for its simplicity, lightweight interface and being coded in C++.

Relative to the Capstone side of things we all worked together on creating this report, individual sections, and presentation where we demoed the basics of our algorithm and email capabilities. I also personally created the rough draft of our Expo Poster. I wanted our design to stand out. After talking to our TA Andrew about hard requirements we are awaiting feedback from McGrath to adjust the style, depth, and visual clarity of our poster.

5.3.2 Whats left to do

My focus for the rest of the term is to incorporate Email and aid in implementation of core DSSE elements. Starting with the POP3 protocol by adding its functionality in place of our demo code for adding/updating to the local database. After talking to our client, we understand that our primary focus should be on this core DSSE algorithm. Implementation of Cloud and Email platforms are not the key components of our project but are real-world applications that help to ground this research project. The final, and arguably most important section, is that of optimization and benchmarking. To create an accurate representation of our DSSE algorithm we are going to compare it side by side to that of our client Attila Yavuzs Bit Matrix Algorithm and another implementation of the DSSE Clusion coded in Java. These benchmarks will act as the analysis and conclusion to our poster and give us specific results that we can show at expo.

5.3.3 Problems impeding progress/Solutions:

The biggest problem this term has been communication with our client and TA. Not being able to meet with them until later in the term it has impaired our ability to gain feedback and understanding of project requirements. We still our initial Gantt Chart to compare progress too and used that until meeting with our client. Another problem that Ive experienced this term is a lack of hard deadlines. In the fall term weve had deadlines for every document. From requirements to specifications these have all been created for our capstone portfolio. However, in this term there has been a very hands-off approach from the capstone team. This is the first time I have worked on such a style of project. It is very realistic to how real-world companies divvying out tasks, so I am grateful that we can practice these self-motivational skills in a less stressful environment when our jobs are not on the line.

In additional, being a research project, it is harder to put into words the progress that weve made outside of project code. For those teams focused on a more implementation heavy projects it is easier to show progress week by week. For example, one week I spent hours looking over and comparing POP3 algorithms against one another to find one that works well without specific implementation. I have listed these in my OneNote as progress, but it is hard to put into

progress without code pushes on GitHub. It is directly relevant to our project and vital to the overall success of meeting our clients specifications.

5.3.4 Looking back and looking ahead:

Overall, I feel that this term has been slower in terms of progress than our original Gantt Chart intended. We will continue to work on the core implementation throughout the rest of the term to have a working state that we can start doing benchmarks against other algorithms. We had a shaky start to the term missing our direct contact with client and TA. We are hopeful that we will can complete all requirements by our client and start the spring term with a satisfactory product that we can improve upon by creating real-world applications to test on for expo.

With implementation of the core DSSE next term should be focused specifically on preparing for expo. Practicing pitches to different audiences, revising the poster within regulation, exploring ways of demoing our research project to appease a general audience and implementing Email and Cloud integration as time permits. This project has been successful this term and our group hopes to finish strong to be able to deliver the product that our client expects of us.

6 RETROSPECTIVE / CONCLUSION

What Went Well?	**What Didn't Go Well?**	**What Can We Change?**
Continuing progress on the project	The term started off a bit slow	Maintain focus on work that needs to get done
Sections of the core DSSE and POP3 compile are are working	Meeting with client and TA didn't start happening until midway through the term	Plan meetings out before the term starts

Last term consisted mainly of planning and organizing for our project. As we headed into this term the tasks for the project were separated into sections as an attempt to parallelize the work being done. With more recent meetings with our client it has become apparent that the main focus of the project needs a slight adjustment. Attila would like to have priority go to creating the DSSE scheme (including update, delete, and the lvl2 pointers). With the refocusing of the project goals some of the tasks given to group members may have to change. Our proof of concept examples (Amazon cloud, Email, and "Dummy" server) are viewed to be more of stretch goal. So far this term, we have been successful in implementing the foundation of the DSSE scheme by importing libraries, creating methods to initialize the database, add to the database, tokenization of text files, and integration of POP3 email. libraries. This progress has gone well but, in light of recent changes we may want to have members double up on implementing the remainder of the core DSSE scheme (Update, Delete, lvl2 pointers, and persistence to disc) and fore go some of the other features. Discussion with our group's TA has also touched on this topic and we may need to plan for some edits to the requirements document as well. Aside from these changes the project is moving along and the group, as a whole seem to be in a good place.

REFERENCES

[1] D. Cash, J. Jaeger, S. Jarecki, C. Jutla, H. Krawczyk, M.-C. Rou, and M. Steiner, "Dynamic searchable encryption in very-large databases: Data structures and implementation," Cryptology ePrint Archive, Report 2014/853, 2014. [Online]. Available: https://eprint.iacr.org/2014/853