

Programming Project IV: Distributed Security

Start Assignment

- Due Apr 20 by 11:59pm
- Points 100
- Submitting a media recording or a file upload
- File Types pdf, py, and txt
- Available after Mar 10 at 12am

The project writeup is available here: [Project 4.pdf \(https://gatech.instructure.com/courses/451732/files/57630061?wrap=1\)](https://gatech.instructure.com/courses/451732/files/57630061?wrap=1)

The project VM can be built by the student:

<https://ubuntu.com/download/desktop> (https://releases.ubuntu.com/23.04/ubuntu-23.04-desktop-amd64.iso?_ga=2.197549867.581489219.1692542523-1257759292.1674584858)

[.\(https://ubuntu.com/download/desktop\)](https://ubuntu.com/download/desktop)

You can follow this guide:

[Ubuntu-Installation.pdf \(https://gatech.instructure.com/courses/333734/files/41777013?wrap=1\)](https://gatech.instructure.com/courses/333734/files/41777013?wrap=1)

[.\(https://gatech.instructure.com/courses/333734/files/41777037/download?download_frd=1\)](https://gatech.instructure.com/courses/333734/files/41777037/download?download_frd=1)

There will be a few packages that you will need, please follow the details from the README file at <https://github.com/TA-gatech/Project4>.

Then download and extract this folder into your Desktop folder in the VM:

```
git clone https://github.com/TA-gatech/Project4.git
```

If you have any questions please join the TA office hours or post them to Ed Discussion

Ed Discussion also has links and mirror links to a pre-installed VM.

[https://gtvault-](https://gtvault-my.sharepoint.com/:u:/g/personal/chayes42_gatech_edu/ETpmGxCeMvdBj6J2_03u8TEB1ve9MXody36pK1TtFS9V7Q?e=zcS0aY)

[my.sharepoint.com/:u:/g/personal/chayes42_gatech_edu/ETpmGxCeMvdBj6J2_03u8TEB1ve9MXody36pK1TtFS9V7Q?e=zcS0aY](https://gtvault-my.sharepoint.com/:u:/g/personal/chayes42_gatech_edu/ETpmGxCeMvdBj6J2_03u8TEB1ve9MXody36pK1TtFS9V7Q?e=zcS0aY)

Good luck!

P4 Rubric

Criteria	Ratings			Pts
Architectural design details – How mutual authentication is achieved in the current implementation of 3S (1 point) – Details on the cryptographic libraries and functions used to handle secure file storage (2 points) – How the user information and metadata related to documents were stored (2 points)	5 pts Full Marks		0 pts No Marks	5 pts
Login Implementation (Report) Implementation Details of login()	2.5 pts Full Marks	1 pts Partial Marks	0 pts No Marks	2.5 pts
Checkin Implementation (Report) Implementation Details of checkin()	2.5 pts Full Marks	1 pts Partial Marks	0 pts No Marks	2.5 pts
Checkout Implementation (Report) Implementation Details of checkout()	2.5 pts Full Marks	1 pts Partial Marks	0 pts No Marks	2.5 pts
Delete Implementation (Report) Implementation Details of delete()	2.5 pts Full Marks	1 pts Partial Marks	0 pts No Marks	2.5 pts
Grant Implementation (Report) Implementation Details of grant()	2.5 pts Full Marks	1 pts Partial Marks	0 pts No Marks	2.5 pts
Logout Implementation (Report) Implementation Details of logout()	2.5 pts Full Marks	1 pts Partial Marks	0 pts No Marks	2.5 pts
Security Analysis details (Report) Results of the static code analysis and the tools used	5 pts Full Marks	3.5 pts Tools not mentioned Tools not mentioned	1 pts Marks for Attempt	5 pts
Threat Modelling details (Report) Threat Modelling and the threats currently handled by your implementation.	5 pts Full Marks	1 pts Partial Marks for Attempt		5 pts
Login Implementation (Code) Login success with user1 using user1.key. Login success with user2 using user2.key. Login fails with user1 using user2.key. Server Returns a token and a status.	10 pts Full Marks	5 pts Partial Marks	0 pts No Marks	10 pts

Criteria	Ratings			Pts
Checkin Implementation (Code) Checkin 1 file with user 1 with Integrity Checkin 1 file with user 2 with Confidentiality. Try to checkin a file with a duplicate file id with user 1 when user1 is the owner. Try to checkin a file with a duplicate file id with user 2 when user1 is the owner.	15 pts Full Marks	7.5 pts Partial Marks	0 pts No Marks	15 pts
Checkout Implementation (Code) Checkout a file with user1 when user 1 is the owner. Checkout a file with user2 when user 2 is the owner. Try to checkout a file with user 1 when user2 is the owner. Tamper with a file after it was checked in and try to checkout that file (should fail)	15 pts Full Marks	7.5 pts Partial Marks	0 pts No Marks	15 pts
Grant Implementation (Code) Grant checkin and checkout and test for access within and after the time limit.	15 pts Full Marks	7.5 pts Partial Marks	0 pts No Marks	15 pts
Delete Implementation (Code) Verify Owner Can Delete and Verify that a user who is not the owner can't delete a file	10 pts Full Marks	5 pts Partial Marks	0 pts No Marks	10 pts
Logout Implementation (Code) Verify the user can logout and the code completes all the tasks.	5 pts Full Marks	2.5 pts Partial Marks	0 pts No Marks	5 pts
Total Points: 100				