CS 1653: Applied Cryptography and Network Security
Spring 2016

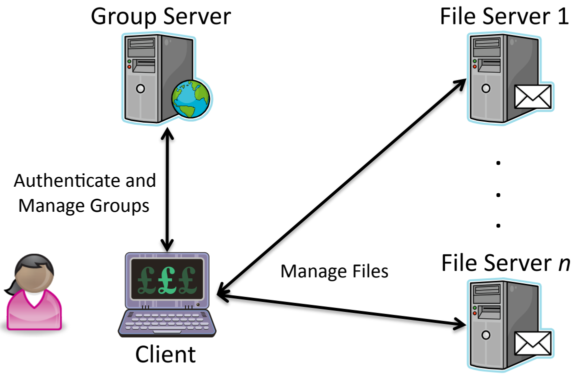
Term Project, Phase 1

Assigned: Tuesday, January 12 **Due:** Tuesday, January 26 11:59 PM

---

1 Background

Over the course of this semester, we will experiment with and apply the security concepts that are covered in lecture by developing a group-based file sharing application that is secure against a number of different types of security threats. At a high level, our system will consist of three main components: a single group server, a collection of file servers, and some number of clients.



The *group server* manages the users in the system and keeps track of the groups to which each user belongs. Any number of *file servers* can be deployed throughout the network, and will rely on the group server to provide each legitimate user with an authentication and authorization token that answers the question *"Who are you, and what are you permitted to do?"* Users within the system make use of a networked *client application* to log in to the system and manage their groups (via the group server), as well as upload, download, modify, and delete files stored in the system (via the file servers).

Your project submissions will be made using the git distributed version control system. You are encouraged to use this system to help coordinate your group work, and follow the git best practice: "commit early and often."

2 What do I need to do?

Phase 1 of the course project is designed to get you thinking about the project at a high level. The above description of the system is deliberately underspecified so that you have the intellectual freedom to consider many different possibilities for how such a system should work. In this phase of the project, you will do just that.

Task 1: Group Formation

Successfully completing the later phases of this project will require that a considerable amount of time and energy be spent exploring and analyzing system design issues, designing security solutions, and implementing/testing your system. To minimize the burden that this places on each of you and to develop your collaborative development skills, all phases of the project—including this one—will be carried out in groups of 2–3 students. Under no circumstance will smaller or larger groups be permitted.

For your first task, you must find 1–2 other students with whom to work for the remainder of the semester. Put some thought into your group choice, as your grades will depend on one another's commitment to your joint work. That being said, all group members will have the opportunity to rate one another's performance on each stage of the project.

Task 2: Security Requirements

After forming your project group, work with your fellow group members to brainstorm a list of *security requirements* that you feel should be respected by a group-based file sharing application like that described above. You may assume that the system will support the following types of operations:

- Create/delete user
- Create/delete group
- Add/remove user u to/from group g
- Upload/overwrite file f to be shared with members of group g
- Download file f
- Delete file f

Given this extremely (and purposely) high-level description of the system's functionality, your group should develop a list of properties that a *secure* group-based file sharing application must respect. For each property, come up with (i) a name for the property, (ii) a definition of what this property entails, (iii) a short description of why this property is important, and (iv) any assumptions upon which this property depends. As an example, consider the following:

> **Property 1: Correctness.** Correctness states that if file f is shared with members of group g, then only members of group g should be able to read, modify, delete, or see the existence of f. Without this requirement, any user could access any file, which is contrary to the notion of group-based file sharing.

The goal of this exercise is to get your group thinking about some the challenges involved with building secure distributed systems. We neither assume that you are file sharing experts, nor that you have prior experience developing secure applications. To begin with, spend some time thinking about what would make *you* trust the security of such a system, and use this intuition to formulate your initial requirements. This will provide you with a starting point that can be refined by examining the features afforded by other systems and reading over relevant portions of your textbook.

Task 3: Setting Up git and Bitbucket

You will submit your projects using the git version control system, using repositories hosted on Bitbucket. Although this phase of the project is a writeup and not code, you will use it as an opportunity to familiarize yourself with git. We will discuss only the very basics of these tools in class. Thus, as part of this phase of the project, you may need to utilize online resources available for these tools.

Begin this task by familiarizing yourself with the concepts of version control in general, and git in particular. One great resource for this is the Pro Git book by Scott Chacon and Ben Straub (`https://git-scm.com/book/en/v2/`), especially chapters 1.1, 1.3, and 2.1–2.4. Note that you do not need to be an expert in git, but you do need to understand its use cases, including why it is an appropriate tool for this project. Once you are comfortable with the concepts, you should install the git client on your local machine using `https://git-scm.com/`

Your git repositories will be stored on Bitbucket, an online host for git repositories. Each member of your group should sign up for Bitbucket, using your `pitt.edu` email addresses to ensure you receive a free unlimited academic account. Optionally, you may install Bitbucket's graphical git client, SourceTree: `https://www.sourcetreeapp.com/`

If you'd like an introduction to the Bitbucket service, visit the Bitbucket tutorial here:

`https://confluence.atlassian.com/bitbucket/`
`git-tutorial-keep-track-of-your-space-station-locations-759857287.html`

A template for your writeup is available to you via an existing Bitbucket repository. As specified in the following section, one member of your group will *fork* this repository, creating your own copy of it under their own account. This fork will be shared with the other group member(s), as well as the instructor.

3 What should I turn in?

The primary deliverable for this phase of your project is a written report that documents your group's activities. This report should have the following structure, as laid out in the provided template:

- **Group information.** List the full name and Pitt email address of each group member.

- **Section 1: Security Properties.** This section should describe the requirements that your group has identified as being relevant to the group-based file sharing scenario. You should aim to find at least 15–20 such requirements, that together will cover at least two different sets of reasonable system assumptions (i.e., threat models). This section should be arranged as a bulleted list of properties that may apply to a file sharing system.

- **Section 2: Threat Models.** This section should describe several sets of trust assumptions that could be made regarding the players in the system. Describe several scenarios in which you expect the file sharing system to be used, and describe the ways in which the various entities with access to the system will be trusted to behave. This section should be arranged as follows:

  1. A paragraph describing a system model: an environment in which you envision your application being deployed.
  2. A paragraph describing the trust assumptions that you would make regarding the players in the system, within this particular system model.
  3. A bulleted list of relevant security properties from Section 1, each with a sentence or two discussing how it applies to this system / threat model. Note that not all of the security properties you define will necessarily be relevant to all of your threat models.
  4. Repeat items 1–3 as needed for additional system / threat models.

  This phase of the project will be graded for content, rather than presentation (hence the bulleted list format); the goal of this exercise is to get you thinking about functionality and security.

- **Section 3: References.** If any of the requirements in Section 1, or any of the system models in Section 2, were inspired by material from books, papers, articles, or existing products, your sources should be cited here.

Your reports should be formatted as HTML, following the template available in the following Bitbucket repository.

`https://bitbucket.org/cs1653-2016/cs1653-project-phase1-init`

To *fork* this repository (i.e., create your own copy of it to work in), **one member** of your group should follow this procedure:

1. Visit the Bitbucket link above and click "Fork" in the menu indicated by the ⋯ symbol.

2. Name your repository using the Pitt usernames of each group member, sorted alphabetically, in the following format: `cs1653-project-phase1-abc1-def2`

3. **Check the box labeled, "This is a private repository."** Failure to do so may be interpreted as cheating!

4. In the pull-down menu labeled, “Forking,” select the option, “Allow only private forks.”

5. Click “Fork repository”.

6. Once you have your own repository forked from the provided code, click “Share,” and add your other group member(s) and the instructor (username `banaanbill`). You should give Bill only read access, but all group members should have Admin access.

   Note: Since your groups are small, you are not required to create a Bitbucket team. However, if you have explored the documentation and discovered teams, and would like to create a team for your project group, you are allowed to do so and fork the repository using this team. However, it will be your group’s responsibility to properly administer this team and ensure the accesses specified above are set properly.

Once you have your own repository, each group member can *clone* it to their own machine to edit the provided HTML file. **Modify this file only within the designated areas.**

You are encouraged to use this version control repository regularly while collaborating on this project, as each individual’s contribution to the group’s work will be judged in part by the version control logs. In addition, *each student in your group* should send an email to `bill@cs.pitt.edu` that indicates his or her assessment of each group member’s contribution to this phase of the project (e.g., *Bob did 40% of the work, and Mary did 60% of the work*).

Your project is due at 11:59 PM on Tuesday, January 26. We will clone your repository immediate after the due date, so you will be graded on whatever changes have been committed and pushed to your repository by this time. Make sure your repository is shared correctly with the instructor well in advance!