CS 1653: Applied Cryptography and Network Security
Fall 2022

Term Project, Phase 4

Assigned: Fri, Nov 11 **Due:** Fri, Dec 02 11:59 PM

---

1 Background

In this phase of the project, you will further extend your file sharing applications to protect against a few more classes of security vulnerabilities. You will be provided with a threat model describing the types of assumptions that you will be able to make regarding the principals in the system, and will be given a list of specific classes of threats for which your system must provide protections. Your deliverables for this phase of the project will again include (i) a brief writeup describing your proposed protections for each type of threat, as well as a description of why these protections are sufficient, and (ii) a set of modified file sharing applications that implement each of your protections.

2 Trust Model

In this phase of the project, we are going to focus on implementing another set of the *security features* required of our trustworthy file sharing service. Prior to describing the specific threats for which you must provide protections, we now characterize the behavior of the four classes of principals that may be present in our system:

- **Group Server** The group server is entirely trustworthy. In this phase of the project, this means that the group server will only issue tokens to *properly authenticated* clients and will properly enforce the constraints on group creation, deletion, and management specified in previous phases of the project. The group server is *not* assumed to share secrets with the file servers in the system.
- **File Servers** In this phase of the project, file servers will be assumed to be largely untrusted. In particular, file servers might leak files to unauthorized users or attempt to steal user tokens.
- **Clients** We will assume that clients are not trustworthy. Specifically, clients may attempt to obtain tokens that belong to other users and/or modify the tokens issued to them by the group server to acquire additional permissions.
- **Other Principals** You should assume that *all* communications in the system might be intercepted by a *active attacker* that can insert, reorder, replay, or modify messages.

3 Threats to Protect Against

Given the above trust model, we must now consider certain classes of threats that were not addressed in the previous phases of the project. In particular, your group must develop defenses against the following classes of threats in this phase of the project:

T5 Message Reorder, Replay, or Modification After connecting to a properly authenticated group server or file server, the messages sent between the user and the server might be reordered, saved for later replay, or otherwise modified by an active attacker. You must provide users and servers with a means of detecting message tampering, reordering, or replay. Upon detecting one of these exceptional conditions, it is permissible to terminate the client/server connection.

T6 File Leakage Since file servers are untrusted, files may be leaked from the server to unauthorized principals. You must develop a mechanism for ensuring that files leaked from the server are only readable by members of the appropriate group. As in previous phases of the project, we stress that the group server cannot be expected to know about all file servers to which its users may wish to connect. Further, your proposed mechanism *must* ensure that some level of security is maintained as group memberships change.

T7 Token Theft A file server may "steal" the token used by one of its clients and attempt to pass it off to another user. You must develop a mechanism for ensuring that any stolen tokens are usable only on the server at which the theft took place (and are thus effectively useless, as this rogue file server could simply allow access without checking the token).

Note that you must also address *all* threats from the previous phase of the project. That is, your new functionality should not invalidate previous requirements.

4 What Do I Need To Do?

This phase of the project has two deliverables. The first deliverable is a semi-formal writeup describing the protection mechanisms that your group proposes to implement, and the second is your actual implementation. We now describe both aspects of the project.

4.1 Mechanism Description

As in Phase 3, the first deliverable for this phase of the project will be a short (e.g., 3–5 page) writeup describing the cryptographic mechanisms and protocols that you will implement to address each of the threats identified in Section 3 of this assignment. This writeup should begin with an introductory paragraph or two that broadly surveys the types of cryptographic techniques that your group has decided to use to address threats T5–T7. You should then have one section for each threat, with *each* section containing the following information:

- Begin by describing the threat treated in this section. This may include describing examples of the threat being exploited by an adversary, a short discussion of why this threat is problematic and needs to be addressed, and/or diagrams showing how the threat might manifest in your group's current implementation.
- Next, provide a short description of the mechanism that you chose to implement to protect against this threat. For interactive protocols, include diagrams explaining the messages exchanged between participating principals. (See the notes from Lecture 11 for example diagrams.) Be sure to explain any cryptographic choices that your group makes: What types of algorithms, modes of operation, and/or key lengths did you chose? Why? If shared keys are needed, how are they exchanged? Recall that security is not absolute nor are any tools appropriate for all situations; every component of your design should be intentional and justified relative to the given threat model.
- Lastly, provide a short argument addressing why your proposed mechanism sufficiently addresses this particular threat. This argument should address the correctness of your approach, as well as its overall security. For example, if your mechanism involves a key agreement or key exchange protocol, you should argue that both parties agree on the same key (correctness) and that no other party can figure out the key (security). You do not need a formal proof, but you should convince me that an attacker can no longer exploit each threat.

After completing one section for each threat, conclude with a paragraph or two discussing the interplay between your proposed mechanisms, and commenting on the design process that your group followed, including any extra credit that you did. Did you discuss other ideas that didn't pan out before settling on the above-documented approach? Did you end up designing a really interesting protocol suite that addresses multiple threats at once? Use this space to show off your hard work!

Finally, spend about one paragraph convincing me that your modified protocols still address the threats T1–T4 described in Phase 3 of the project. Full credit for Phase 4 requires that all Phase 3 threats are still protected against.

As in the last phase of the project, 10% of your grade for this phase of the project is based upon approval of your design in a meeting with the instructor before the deadline. As before, this discussion can occur during office hours or by appointment, and I will reserve and announce appointment slots specifically for these discussions. Have a *complete* writeup pushed to GitHub in advance of this meeting to facilitate the conversation, but note that there is no expectation that your writeup is *perfect*. I expect that these conversations will result in changes, but we cannot have a productive discussion if the necessary details are not decided and written down. I will not attempt to evaluate a scheme that is described only out loud or that lacks the necessary details. Remember that, if there are flaws in your approaches, we will not design new approaches in our meeting. This means you may need to meet multiple times, so plan ahead!

4.2 Implementation Requirements

We strongly recommend that you leverage the expertise developed in Homework HW1 and use the BouncyCastle cryptography API to incorporate any cryptographic functionality

that you may need. As before, this project will be graded using the CS Linux Cluster. Please ensure that your code runs correctly under Linux on these machines well before the due date.

5 Extra Credit

As in previous phases of the project, you again have the opportunity to earn up to 8% extra credit. Should you happen to complete the required portions of the project early, consider adding in extra functionality in exchange for a few extra points (and a more interesting project). Any extra features that you add could qualify, so brainstorm as a group and see what you come up with! If you opt to do any extra credit, be sure to include a brief description of it in the discussion section of your writeup.

(Unless a previously-implemented extra credit feature requires substantial updates within the new threat model, you cannot use the same feature for extra credit in multiple phases.)

6 What (and how) do I submit?

Your grade for this project will be based upon your technical writeup (50%), approval of your design by the instructor (10%), a demonstration and assessment of the code that your team produces (35%), and scheduling a demo with the TA prior to the deadline (5%).

An initial writeup skeleton is available to you via the following GitHub repository:

`https://github.com/2231-cs1653/cs1653-project-phase4-writeup`

You should copy the provided HTML file into the documentation directory of your main project repository (`doc/phase4-writeup.htm`). Modify this file *only* within the denoted areas.

Within your project repository (your existing `cs1653-project-*` repository from previous phases), you should include the following files and directories.

- `src/` In this directory, include all of your source code that is needed to compile your project. Please do not commit any JAR or class files, as we will be rebuilding your code before we evaluate it (and it is common version-control etiquette not to commit files that can be re-derived from the included source). Also, please do not commit any publicly available libraries (e.g., Apache Commons, BouncyCastle) that you make use of—include instructions for acquiring those libraries in `doc/compile.txt` (see below).
- `doc/` In this directory, include all documentation for your project, including *at least* the files named below.
  - `doc/compile.txt` This text file should include instructions for compiling your code and provide links to publicly available libraries used in your project.
  - `doc/usage.txt` This text file should include instructions on how to use your system. Explain how to start your group server and file server, how to start your client application(s), and how to invoke each of the client applications' supported operations.

– `doc/phase4-writeup.htm` Copy the provided HTML writeup skeleton and edit it *only* where noted, as discussed above.

As mentioned previously, your repository's commit log will serve (in part) to ensure each individual is contributing to the group project. In addition, *each student in your group* should send an email to `bill@cs.pitt.edu` that indicates their assessment of each group member's contribution to this phase of the project.

Your project is due at the precise date and time stated above. We will clone your repository immediately after the due date, so you will be graded on whatever changes have been committed **and pushed** to your repository's main branch by this time. No changes made after this point will be considered in your demo or in grading your project. Make sure you understand the submission process well in advance!