CS 1653: Applied Cryptography and Network Security
Fall 2017

Term Project, Phase 4

Assigned: Tuesday, November 7 **Due:** Sunday, November 26 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.

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, it would be helpful to provide a diagram 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?
- 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).

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 a discussion of your writeup in-person with the instructor.

4.2 Implementation Requirements

We strongly recommend that you leverage the expertise developed in Homework HW2 and use the BouncyCastle cryptography API to incorporate any cryptographic functionality that you may need. As before, this project will be graded using the machines in Sennott Square 6110. Please ensure that your code runs correctly under Linux on these machines before the due date.

5 Extra Credit

As in previous phases of the project, you again have the opportunity to earn up to 5% 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 will 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.

6 What (and how) do I submit?

Your grade for this project will be based upon your technical writeup (40%), your discussion of this writeup with the instructor (10%), a demonstration and assessment of the code that

your team produces (40%), scheduling a demo with the TA prior to submission (5%), and properly submitting your assignment (5%).

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.md` (see below).
- `doc/` In this directory, include all documentation for your project, including *at least* the files named below.
  - `doc/compile.md` This Markdown file should include instructions for compiling your code and provide links to publicly available libraries used in your project.
  - `doc/usage.md` This Markdown 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/p4_extra_credit.md` *(optional)* If you have gone beyond the requirements of the project as described above, please explain what you have done in this Markdown file.
- `reports/phase4-writeup.md` This will be where the Markdown version of the report described above will go. Professor Lee will read directly from your repository during your team's discussion(s) regarding your proposed mechanisms for this phase of the project.

Preparing for submission. When your repository is ready for submission, commit and push your repository back to GitHub. Then, create a git *tag* called `phase_4`:

```
git tag -a phase_4 -m "Phase 4 submission"
```

Then, push this tag to GitHub:

```
git push origin phase_4
```

In git, a tag is simply human-readable shorthand for an (otherwise unreadable) commit hash. The first command, above, links your commit hash to the string `phase_4`, while the second command syncs that tag to GitHub so that others can use it. After this tag is available, anyone with access to your repository can examine your Phase 4 submission by typing `git checkout phase_4` from inside a local clone of your repository.

Submission. Your project is due at 11:59 PM on Sunday, November 26. To timestamp your submission, run the command `git log -n 1` and email the TA (veronica@cs.pitt.edu) the commit hash for your submission **prior to the due date**. After cloning your repository, we will grade the version of your code that corresponds to this commit hash.

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 `adamlee@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*).