IT Security: Practicals

Implementing and Attacking Cryptographic Primitives and Protocols

Sebastian Ramacher,
Philipp Harb, Martin Mandl, Samuel Sprung, Stefan Steinegger

October 3, 2016

Contents

1 Rules

You must perform all tasks in groups of two members.[1] We will provide a Git repository to coordinate both your teamwork and the submission. Using Git is mandatory (see Section 1.1 for details). At the end of the term mandatory group interviews (Abgabegespräche) conclude the exercise. Within the interviews you

- have to be able to explain the theory behind the tasks.
- have to be able to explain your practical work. (Every group member has to know the complete work of the group.)
- have to explain if your Git statistics look abnormal, e.g. only one of the team members committed regularly.
- have the opportunity to inspect and discuss your submission (Einsichtnahme).

The grading is done as follows: $\geq 50\% : 4, \geq 62.5\% : 3, \geq 75\% : 2, \geq 87.5\% : 1$. The percentage is calculated from the sum of the received points divided by all possible points multiplied by a factor depending on the performance at the group interviews. You need to pass the group interview to get a positive grade overall. You will get a grade as soon as you submit something to any of the tasks.

In case you are unable to acquire enough points to pass, you will have the chance to extend the submission to the task where you have submitted a solution but received the least points.

You are *not* allowed to use external code other than provided by the framework. If plagiarism is detected, all involved teams will fail the course.

The preliminary schedule (task presentation, submission deadlines) can be found in Table 1.

Date & Time	**Remark**
2016.10.03 16:00	Registration. Task 0.
2016.10.10 23:59	Registration. Task 0 deadline.
2016.10.10 16:00	Presentation of Task 1.
2016.10.31 23:59	Task 1 submission deadline.
2016.11.07 16:00	Presentation of Task 2.
2016.12.05 23:59	Task 2 submission deadline.
2016.12.12 16:00	Presentation of Task 3.
2017.01.16 23:59	Task 3 submission deadline.
2017.01.23–2017.01.27	Group interviews

Table 1: Course schedule.

All tasks ask you to answer questions for an assignment document. The goal of the assignment documents is to reinforce the knowledge on the particular topic. By properly

[1] If you are unable to find a partner, we will assign one to you.

answering the question, you will already have reached the educational objective. Therefore, we will not check the answers for correctness. We will only check if the assignment was done.

The questions from the assignment document will be part of the group interviews.

If the assignment document is missing for a task, you will receive no points for this particular task.

All submission deadlines are hard. Submissions after the deadline are not considered.

If you have any questions, consult one of the following sources for further information:

- Newsgroup: tu-graz.lv.it-sicherheit
- Via Email: itsec-team@iaik.tugraz.at
- (If needed) question times: about one week before the submission deadlines.

1.1 Task Submission

If you not familiar with Git, please have a look at the vast amount of resources about Git online. A book called Pro Git is freely available online.

All tasks are submitted using your group's Git repository and need to be tagged. Each proper submission must be tagged with a Git *tag* called

- `submission-1` for Task 1.
- `submission-2` for Task 2.
- `submission-3` for Task 3.

To correctly tag the currently checked out commit and push the new tag to the server run:

```
% git tag submission-X
% git push origin master submission-X
```

You can easily verify your submission by cloning a fresh copy of your repository into a new directory and trying to check out the corresponding tag. The Git commands used by us to clone your submissions are semantically equivalent to:

```
% git clone git@teaching.student.iaik.tugraz.at:its16g0XX.git
% cd its16g0XX
% git checkout submission-X
```

☞

You might accidentally tag the wrong commit for submission. To re-submit another commit simply create a new tag named `submission-X-Y` where `Y` is an increasing number. Assuming that you want re-submit for the first time, run the following commands:

```
% git tag submission-X-1
% git push origin master submission-X-1
```

To re-submit again use `submission-X-2`, `submission-X-3` and so on.

The tags that qualify for submission, i.e. were created before the deadline, and reference a commit before the deadline, are sorted according to their name, and we will consider the tag that sorts last as your submission.

We will checkout your submission shortly after the deadline has passed, and we will publish a newsgroup posting in tu-graz.lv.it-sicherheitlisting the received Git commit IDs that are considered to be your final submissions for the tasks.

⚠

Any discrepancies between the commit IDs published by us in the newsgroup and the actual tags in your repositories will be resolved in favor of the published commit IDs - Attempts to (re-)tag your submissions after the deadline will be penalized.

⚠

Never force push to the repository.

The directory structure of your repository should look similar to:

```
assignment-document/ - your assignment documents for all tasks
tls/                 - framework and your solutions for Task 1-2
blockchain/          - framework and your solutions for Task 3
```

All your submissions have to include

- the full source code, and
- result files where applicable.

⚠

Keep your repository clean from build artifacts like object files or executables that were generated during the build process. The same applies for generated doxygen documentation and other by-products of the build process.

1.2 First Steps in the Git Repository

After you have received your Git repository, perform the following steps to get started:

1. Update author information:

   ```
   % git config --global user.name "John Doe"
   % git config --global user.email "jd@student.tugraz.at"
   ```

2. Ignore build artifacts:

   ```
   % echo build/ >> .gitignore
   % git add .gitignore
   % git commit -m "Ignore build artifacts"
   ```

2 Task 0: Registration

⚠

HARD SUBMISSION DEADLINE: 2016.10.10 23:59

In order to fully sign up for this course, each group has to register in STicS. The first member registers a group and invites the second member to the group. Make sure select KU as topic.

References