Assessed Paired Coursework 2 — Broadcast Message Secure Reader

Learning Outcomes

- *Practical experience of analysing, designing, implementing and validating solutions to computer network security challenges using common network security tools and formal methods.*
- *Ability to deal with complex issues and make informed judgements about network security in the absence of complete or consistent data.*
- *Exercise substantial autonomy and initiative in addressing computer network security challenges.*
- *Showing initiative and team working skills in shared computer network security application development.*
- *Demonstrate critical reflection on network security issues.*

1 Overview

This assessed coursework is for MSc students taking F21CN. It is worth 30% of the overall course mark for Computer Network Security. It is one of two pieces of assessed coursework for this course. This coursework is an exercise in creating, using and revoking X.509 and PGP certificates. It involves developing an application that can be securely be used to read broadcast messages digitally. The application is capable of reading messages from a set of trusted authors (using PGP certificates). The application is certified by a local Certification Authority (CA). Context of use: the application is to be distributed to the members of an organisation to read messages authored by a selected number of members.
The choice of programming language to implement this application is left to the pair. You can choose between Java and Python. If you want to use another programming language, please get agreement from the lecturer first. Indicate your choice to the lecturer. The learning objective of this coursework is for you to become familiar with the concepts of *certificates and signatures*. The work should be done in pairs. However, pairs of students also have to join together with other pairs to form a wider group of people who are prepared to sign each other's certificates. Students having difficulties to find partner should contact your lecturer. It is recommended that the pairs do their collaborative work using the University and MACS systems: Teams, Word Online, GitLab Student[1].

2 Pair Tasks

Each pair should perform the following tasks:

1. create one self-signed PGP certificates and private keys for each of member of the pair
2. create one self-signed X.509 certificate and private key for the pair
3. create a local CA run by the pair (the local CA should be given a suitable X.500 name and have a self-signed X.509 certificate created for it; it may be appropriate to take steps to ensure that this certificate has the basic constraint extension set on it to identify it as a CA certificate)
4. form a group with at least one other pair of students and do group activities:

[1] http://gitlab-student.macs.hw.ac.uk/

   (a) exercise due diligence in using key to sign other pairs' certificates using your local CA
   (b) get your pair's certificate signed by at least one other pairs' local CA

5. with the wider group of students hold *virtual* key party(ies) for members to sign each other's OpenPGP certificates

6. each member of the pair, encrypt a plaintext message[2] with your PGP private key and share with wider group

7. write an application to read messages written by authors of which you were able to sign OpenPGP certificates

8. sign the application with the private key corresponding to the pair's X.509 certificate

9. demonstrate your application works correctly using a recorded video[3] with screen sharing involving both members of the pair (maximum length: 5 minutes)

10. consider now that one of your pair's PGP certificate has been compromised as well as your pair's X.509 certificate and perform the followings:

    (a) create a revocation certificate for the compromised PGP key and share the revocation certificate with the wider group
    (b) create a certificate revocation list including your pair's X.509 certificate
    (c) once you have received at least one PGP revocation for a public key your application uses, create a newer version of your application; for this, you will need to issue a new X.509 certificate

11. submit pair and individual reports describing your work by the due date (see Section 3).

X.509 certificates should have a sensible X.500 name. PGP certificates should have sensible identifiers of your owner and include at least an e-mail address and a small photograph of them. Students should exercise due diligence in key parties when signing each other's PGP certificates. The application should enable a user to select a message in the local file system, and display the name of the author of the message and the plain text message. In case a message was encrypted with a revoked key, the application should warn about it and should not display the message (pairs can decide to demonstrate or not demonstrate this feature in their demonstration video).

3 Reports

A **pair report** (up to 8 pages) should be jointly[4] written and submitted, it should:

1. provide the URL to the demonstration recording[3], and either the URL to your code joint GitLab Student[5] project or the source code as appendix to the report

[2] Use respectful plaintext messages, if you do not have a message at hand, you can pick your favourite quote from NCSC's email security guidance: `https://www.ncsc.gov.uk/collection/email-security-and-anti-spoofing`

[3] The recording should be made available on Microsoft Stream (Teams recorded meetings are automatically uploaded to Microsoft Stream) and be shared with the lecturer but keep the video private / do **not** tick "Allow everyone in your company to view this video".

[4] Marks will be given based on each pair's demonstration and written submissions. Pair members may also elect to be individually assessed, but need to inform the lecturer at least two weeks before the deadline.

[5] `http://gitlab-student.macs.hw.ac.uk/`

2. succinctly describe the project — what your pair did and what you produced, include an introduction section, discussing what you expect to learn from the assignment in general (and for each task), and describe the environment that you used to complete the tasks (e.g., what machines, software and versions)
3. list certificates, source files and code along with a brief account of how it works (prior and after revocations), use either screenshots or just cut-and-paste the command line with the responses, documenting the steps taken on each of the tasks above
4. explain any observations that are interesting or surprising, document any difficulties that you met while doing any of the tasks.

An **individual report** (up to 2 pages) should be individually written and submitted, it should:

1. include an account of who did what on your pair work, give a percentage estimate
2. critically discuss the proposed security solution in terms of its security policy, threat model and a risk assessment of how well the deployed security measures mitigate threats[6]
3. in particular, discuss the impact of performing these activities *virtually*.

4 Note on plagiarism and collusion

This is a group coursework and you are expected to work in pairs to complete the coursework tasks. Your coursework submissions will be automatically checked for plagiarism. Here are some further points to take into consideration (here, *your* refers to the pair of students in the group):

- Coursework reports must be written in your own words and any code in your coursework must be your own code. If some text or code in the coursework has been taken from other sources, these sources must be properly referenced.
- Failure to reference work that has been obtained from other sources or to copy the words and/or code of others is plagiarism and if detected, this will be reported to the School's Discipline Committee. If a student is found guilty of plagiarism, the penalty could involve voiding the course.
- Students must never give hard or soft copies of their coursework reports or code to others. Students must always refuse any request from others for a copy of their report and/or code.
- Sharing a coursework report and/or code with other students is collusion, and if detected, this will be reported to the School's Discipline Committee. If found guilty of collusion, the penalty could involve voiding the course.
- And remember: the consequences of taking unacceptable short cuts in coursework are much worse than getting a bad mark (or even no marks) on a piece of coursework. There has been one case this year where a student was awarded on Ordinary degree (rather than an Honours degree) because of the sanction imposed by the University's Discipline Committee. The offence was plagiarism of coursework.
- Further information on academic misconduct can be found in: https://www.hw.ac.uk/students/doc/discguidelines.pdf

[6] Note that this last item differs in the assessment of F20CN and F21CN. Pairs may be composed of F20CN and F21CN students.

5 Submission

The written reports must be submitted on Vision (submission links on the Vision page for F21CN). Each report must be submitted as a single file on Vision Turnitin. Include a summary/conclusion section, where you discuss whether your expectations were met, highlighting issues of particular importance, what you learned, and suggesting further work.

Your coursework is due to be submitted by 3:30pm on Tuesday 1st of December, 2020.

The course applies the University's coursework policy.

- No individual extension for coursework submissions.
- Deduction of 30% from the mark awarded for up to 5 working days late submission.
- Submission more than 5 working days late will not get a mark.
- If you have mitigating circumstances for an extension, talk to your Personal Tutor and submit a Mitigating Circumstances (MC) form online[7].

You should expect feedback on your submitted coursework by Tuesday 22nd of December, 2020.

6 Marking Scheme

Total marks for F21CN Coursework 2: 100

1. *Certificate, message signing, message reading, revocations*
   These should conform to the specification and be detailed and evidenced in the report. (25 marks)
2. *Application code*
   The code should be commented, (snippets) presented in the report and demonstrated. The application functions and security implementation must be evidenced in the report and in the demonstration recording. (25 marks)
3. (individual part) *Security analysis, threat model, risk assessment*
   The security norms at stake should be critically discussed, it should discuss the threat model considered and give a detailed risk assessment. (25 marks)
4. *Report and demonstration*
   The report should be well structured and provide the necessary codes, commands and screenshot to document the work done (see Section 3). The students should demonstrate and explain in a recording the steps to prepare the application, and to read messages. (25 marks)

[7] http://www.hw.ac.uk/students/studies/examinations/mitigating-circumstances.htm