Assessed Paired Coursework 2 — Trusted Documents Broadcasting

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 25% of the overall course mark for Computer Network Security. It is one of two pieces of assessed coursework for this course. This is a group assessment where each group is expected to be a pair of students. Each pair has to be composed of students from the same campus. The pair can be from the same or different versions of the course: F20CN/F21CN.

This coursework is an exercise in creating and using digests, and X.509 and PGP certificates. It involves developing an application that can be used to verify the authenticity of broadcasted digital documents. The application is capable of reading a document and it signature and to check that it is originating from a set of trusted authors (using PGP certificates). The list of trusted authors is certified by a local Certification Authority (CA). Context of use: the application is to be distributed to the members of an organisation to guarantee that only documents authored by a selected number of members are to be accepted by the wider membership of the organisation. The application should be implemented as a commandline client/server application with the broadcasting server distributing list of trusted authors and documents, and the client verifying authenticity of the server's communications and the author's signature of each broadcasted document. The application can also be implemented with no network interface and therefor working on the commandline locally only, see Section 6 for the implication of not including a network interface to the application. You are expected to add and document your own extra features such as for example managing update to documents, or handling revocations, or GUI.
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. 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 the campus lecturer. It is recommended that the pairs do their collaborative work using the University and MACS systems: Teams, Word Online, GitLab Student[1].

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

2 Tasks

Certificates and Signatures

Each member of the pair should perform the following tasks:

(i) Create one self-signed PGP certificate[2] and private key.

(ii) With the wider group of students, hold on campus or virtual key party(ies)[3] to sign each other's PGP certificates.

(iii) Create a plain text document[4]; sign the document using your private PGP key; share with the wider group of students your PGP certificate, your document and its signature.

Each pair should perform the following tasks:

(iv) Create a new X.509 certificate and private key for your pair.

(v) 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).

(vi) Form a group with at least one other pair of students and do these group activities:

  (a) Exercise due diligence in using key to sign other pairs' certificate using your local CA.

  (b) Get your pair's certificate signed by at least one other pairs' local CA.

Application Development

Each pair should perform the following tasks:

(1) Develop your application. The first component of the application (`server`) can sign and provide a list of PGP certificates of trusted authors using the pair's X.509 private key (`broadcastlist`), and can provide a document and its signature which was created by the author with its PGP private key (`broadcastdoc`). The second component of the application (`client`) can receive a list of trusted authors's PGP certificates and verify the authenticity of the list using the pair's X.509 certificate (`receivechecklist`) and can receive a broadcasted document and its signature and verify the authenticity the signature checking that the document was signed by an author among the list of trusted authors (`receivecheckdoc`). The components should first be implemented as two command line tools taking the operation name and name of local files as arguments.

(2) You can test you application using documents and signatures shared in task (iii). Make sure to test both normal functions (positive testing) and error cases (negative testing).

(3) You can then extend your implementation to include network interface.

[2] PGP certificates should have sensible identifiers of your owner and include at least an e-mail address and a small photograph of them.

[3] Students should exercise due diligence in key parties when signing each other's PGP certificates.

[4] Use respectful text in your document, if you do not have a document 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

(4) Once you have a working application with network interface, you can consider adding extra features.

(5) Demonstrate your application works correctly using a recorded video. Submit pair's report, source code and demonstration, and individual reports (see Section 3).

3 Reports and Demonstration Recording

A **pair report** (up to 8 pages) should be jointly[5] written and submitted, it should:

1. 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).
2. List certificates, source files and code along with a brief account of how it works, use either screenshots or just cut-and-paste the command line with the responses, documenting the steps taken on each of the tasks above.
3. 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 partially or fully online.

A **pair demonstration recording** (up to 5 minutes), it should:

- involve both members of the pair,
- be a screencast,
- demonstrate the use of the application,
- explain the main elements of the source code of the application.

For the pair demonstration, we recommend you use Teams to record a meeting where you would share your screen(s). Such recording is then available on Microsoft Stream where you can do simple trimming if necessary. You can then download locally the recording to upload it on Canvas.

[5] 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.

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

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 at:
  https://www.hw.ac.uk/uk/students/studies/examinations/plagiarism.htm

5 Submission

The written reports, demonstration recording, URL to the pair's GitLab Student project must be submitted on Canvas. Each report must be submitted as a single file. 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 on Wednesday 30th of November, 2022 (local times: 3:30pm Edinburgh / 11:59pm GA / 11:59pm Dubai).

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 Wednesday 21st of December 2022.

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

6 Marking Scheme

Total marks for F21CN Coursework 2: 100

1. Certificates, document signing, verifications
   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 screen-shot to document the work done. (25 marks)

Grade guidance

- A 70% and over Full implementation of the specification including network interface and extra features. Excellent quality of reports, demonstration and code.
- B 60-69% Full implementation of the specifications including network interface but without necessary extra features. Very good quality of report, demonstration and code.
- C 50-59% Implementation of most of the specifications with partial network interface, without extra features. Good quality of report, demonstration and code.
- D 40-49% Partial implementation of the specifications without network interface, without extra features. Acceptable quality of report, demonstration and code.
- E 30-39% Partial implementation of the specifications without network interface, without extra features. Weak report, demonstration and code.
- F 0-29% Limited implementation of the specifications without network interface, without extra features. Incomplete report, demonstration and code.