Security and Cryptography
Item 1 - Security Protocol Report (CW)

Module Coordinator	Dr David Williams <david.m.williams@port.ac.uk>
Issued	March 2025
Code	M22359/M33122 Security and Cryptography
Purpose	This document specifies the security protocol analysis to be completed for the security protocol component of the module to be submitted as an assignment worth 50% of the overall module mark.
Version	1.0 — Initial Release

Schedule & Deliverables

Item	Value	Format	Deadline
Item 1 - Report (CW)	50%	PDF	2025-05-19 16:00 **BST**

Notes

- Submit your answers to the following exercises in one single PDF file to the submission dropbox on moodle.
- Use the accompanying examples to help you complete the security protocol analysis; together.
- This is individual work. You *will* learn extensively from each other, but any work you submit must be your own.
- The Extenuating Circumstances procedure is there to support you if you have had any circumstances (problems) that have been serious or significant enough to prevent you from attending, completing or submitting an assessment on time. If you complete an Extenuating Circumstances Form (ECF) for this assessment, it is important that you use the correct module code, item number and deadline (not the late deadline) given above.
- ASDAC are available to any students who disclose a disability or require additional support for their academic studies with a good set of resources on the ASDAC moodle site
- The University takes any form of academic misconduct (such as plagiarism or cheating) seriously, so please make sure your work is your own. Please ensure you adhere to our Student Conduct Policy and watch the video on Plagiarism.
- Any material submitted that does not meet format or submission guidelines or falls outside of the submission deadline could be subject to a cap on your overall result or disqualification entirely.
- If you need additional assistance, you can ask your personal tutor, student engagement officer ana.baker@port.ac.uk, academic tutor eleni.noussi@port.ac.uk or your lecturers.
- If you are concerned about your mental well-being, please contact our Well-being service

SECRYPT - Security and Cryptography - Exercises

Use the accompanying examples to help you complete the following exercises; together, the examples and exercises help develop a technical understanding of a broad range of ciphers.

Make a copy of the Item 1 Template file and populate it with your student number and answers to each of the following exercises; do not change the formatting.

Print to PDF and submit this single PDF file to the submission dropbox on moodle.

Exercise 1 - Elliptic Curve Diffie Hellman Key Exchange

For this exercise you will calculate shared secrets established between protocol participants based on values generated using the functions `generate_DH_values` and `generate_ECDH_values` in `protocol_generator.py`

This function takes a three digit number `num` as input and returns a set of parameters.

To generate your parameters enter the last three digits of your student number.

Here's an example of using the python function to generate the plaintext:

```
>>> generate_ECDH_values('123')
```

a. Calculate the secret that Alice and Bob agree upon via Elliptic Curve Diffie Hellman Key Exchange using the values generated using `generate_ECDH_values`. Also provide the coordinates sent over the public channel by Alice and Bob in establishing the shared secret.

[15 marks]
5 marks for correct coordinates of shared secret
and 5 marks for each correct calculation of a sent value

[15 marks total for exercise]

Exercise 2 - Authentication Properties

Consider the following authentication protocol given in common syntax:

```
A, B:      principal
Na, Nb:    nonce
pk, sk:    principal -> key (keypair)

1.  A->B: {A,B,Na}sk(A)
2a. B->A: {Nb,Na}pk(A)
2b. B->A: {Na,A}sk(B)
3.  A->B: {Nb,A}pk(B)
4.  B->A: {Nb,Na}sk(B)
```

We shall call this protocol `PROTOCOL_TWO`

The SPDL specification of `PROTOCOL_TWO` has been written for you, it is to be found in the file `PROTOCOL_TWO.spdl`.

a. In your own words, describe a counter-example that illustrates that the removal of the message 2b causes the protocol to fail to exhibit Weak Agreement from the perspective of the initiator `I`.

[10 marks]

2 marks for each accurate point that aids the description (up to 10 marks max)

b. With reference to `PROTOCOL_TWO`, explain (in your own words) how Weak Agreement and Non-Injective Agreement differ from each other.

[15 marks]

3 marks for each accurate and relevant point made (up to 15 marks max)

[25 marks total for exercise]

Exercise 3 - Authentication Protocol Specification

For this exercise you will analyse a security protocol generated using the function `generate_protocol_three(num)` in `protocol_generator.py`

This function takes a three digit number `num` as input and returns a protocol specification.

To generate your protocol for analysis, enter the last three digits of your student number.

Here's an example of using the python function to generate the plaintext:

```
>>> generate_protocol_three('123')
```

We shall call this protocol `PROTOCOL_THREE`, which is the protocol generated by substituting the last three digits of your student number into the function above.

a. Construct an SPDL specification of `PROTOCOL_THREE` for verification in Scyther.

[10 marks]

10 marks for a correct construction without errors
else 2 marks for each accurate element (up to 8 marks max)

b. To what extent does `PROTOCOL_THREE` mutually authenticate the protocol participants? Justify your answer through security protocol analysis using scyther, including descriptions of relevant counter-examples and comments upon the contribution of specific message components.

[20 marks]

2 marks for each accurate and relevant point made (up to 20 marks max)

[30 marks total for exercise]

Exercise 4 - Authentication Protocol Specification Using Tickets

For this exercise you will analyse a security protocol generated using the function `generate_protocol_four(num)` in `protocol_generator.py`

This function takes a three digit integer `num` as input and returns a protocol specification.

To generate your protocol for analysis, enter the last three digits of your student number.

Here's an example of using the python function to generate the protocol:

```
>>> generate_protocol_four('123')
```

We shall call this protocol `PROTOCOL_FOUR`; it is the protocol generated by inputting the last three digits of your student number into `generate_protocol_four(num)`.

a. In your own words, describe the message exchange of `PROTOCOL_FOUR`.

[15 marks]
3 marks for each accurate point made that aids the description of the protocol and its essential use of tickets (up to 15 marks max)

b. Construct an SPDL specification of `PROTOCOL_FOUR` for verification in Scyther.

[15 marks]
15 marks for a correct construction without errors
else 3 marks for each accurate element (up to 12 marks max)

[30 marks total for exercise]

[100 marks total for assignment]