Chapter 2

Preliminaries and Definitions

In this chapter, we introduce the fundamental elements of an exam system. We begin the treatment with an informal description of roles, principals, and threats, and conclude the chapter with the formal specification of these fundamental elements in the applied \(\pi\)-calculus. In consequence, describing and formalising a specific exam becomes easier at the sole price of further expanding or specifying these general concepts. We anticipate that we view an exam as a protocol that involves various tasks defining roles played by various principals through various phases. Hence, exam, exam protocol, or exam system are used interchangeably. With a security take, an exam is expected to withstand a threat model meeting a number of security requirements.

Outline of the chapter. Section 2.1 discusses the levels of detail and abstraction to characterise tasks. Section 2.2 introduces possible roles for exams and principals that play the exam roles. Section 2.3 identifies the typical phases of an exam. Section 2.4 details the potential security threats associated with the exam roles. Section 2.5 classifies exams by type and category. Section 2.7 outlines the basic constituents of the applied \(\pi\)-calculus. Section 2.8 introduces the formal framework by specifying the formal model of an exam.

2.1 Tasks

The ultimate goal of an exam is to assign a mark to a candidate. A number of tasks occur during an exam to fulfil that goal, such as generating the set of questions, building the tests, and marking them. We observe that the number of tasks cannot be fixed as it may change over two possible dimensions: the level of detail and the level of abstraction for the specification of the exam protocol.

The level of detail establishes whether a task should be explicitly mentioned or not. For example, the task of sealing the tests once they are generated might not be detailed in the specification of an exam. In security protocols, we often prescribe the task of using a nonce in a message, yet omitting the task of fetching it. Experience teaches us that a specification should make very clear (in)security assumptions about the protocol environment, otherwise the analysis may yield
Chapter 2. Preliminaries and Definitions

debatable findings. In this vein, Needham stated that the public-key Needham-Schröder protocol considered the attacker as an outsider but never made this explicit [Nee02], a threat model that would remove the opportunity for Lowe’s attack.

The level of abstraction establishes whether a task should be expanded into sub-tasks or not. For example, the task of generating the set of questions for the exam can be expanded into appointing question setters, setting the guidelines for wording and difficulty, etc. Similarly, in security protocols the task of fetching a nonce may be expanded into accessing a random number generator, running it, and receiving its output securely. These may be further expanded in turn.

From the security analysis standpoint, the levels of detail and of abstraction must be chosen with care in order to limit the necessary assumptions to realism and according to a threat model. For example, the details of generating the set of questions can be abstracted away if question setters are trusted, hence they keep secret the questions until after testing takes place. Conversely, such details should be explicitly described in case of a threat model that considers collusion between question setters and candidate. It follows that a threat model with less reliance on trusted parties normally requires a greater level of detail and abstraction for the specification of an exam than a threat model with more security assumptions. This is demonstrated in the following chapters and especially in Chapter 6, where the presence of trusted parties is progressively reduced in an exam protocol family series.

2.2 Roles and Principals

A role is a set of principals who perform a specific set, possibly of cardinality one, of tasks. During exams, an obvious role is the candidate role, of taking the exam to get a mark that may give the candidate a goal — such as obtaining a qualification, passing a periodical academic assessment, or being selected through a public examination. Of course, the candidate role could be specified, if needed, at a lower level of abstraction, and examples can be derived from actual protocol specifications. Other possible roles, also called authority roles, are as follows

- The registrar role, of checking the eligibility of candidates who wish to take an exam, and of populating a list of registered candidates accordingly.
- The question committee role, of generating and building the tests, passing them to invigilators, forming the test answers in case of multiple-choice tests and passing them to the examiners.
- The moderator role, of setting the guidelines for wording and difficulty of questions, and liaising with the question committee to independently ensure that the tests conform to the guidelines as well as readability and appropriateness standards.
- The invigilator role, of distributing tests to candidates, of checking candidates’ identities, of following candidates while they take their test preventing them from misbehaving.