An Unplugged Didactical Situation on Cryptography between Informatics and Mathematics

Evmorfia-Iro Bartzia, Michael Lodi, Marco Sbaraglia, Simon Modeste, Viviane Durand-Guerrier, Simone Martini

Informatics in Education, 2024, Vol. 23, No. 1, 25–56

DOI: 10.15388/infedu.2024.06

An Unplugged Didactical Situation on Cryptography between Informatics and Mathematics

Evmorfia-Iro BARTZIA[1], Michael LODI[2,3],
Marco SBARAGLIA[2,3], Simon MODESTE[1],
Viviane DURAND-GUERRIER[1], Simone MARTINI[2,3,*]

[1]*Institut Montpelliérain Alexander Grothendieck (IMAG), Université de Montpellier, France*
[2]*Dipartimento di Informatica-Scienza e Ingegneria (DISI), Università di Bologna, Italy*
[3]*Laboratorio Nazionale 'Informatica e Scuola', CINI, Italy*
e-mail: evmorfia-iro.bartzia@umontpellier.fr, michael.lodi@uibo.it,
marco.sbaraglia@unibo.it, simon.modeste@umontpellier.fr,
viviane.durand-guerrier@umontpellier.fr, simone.martini@unibo.it

Received: Match 2023

Abstract. In this paper, we present an activity to introduce the idea of public-key cryptography and to make pre-service STEM teachers explore fundamental informatics and mathematical concepts and methods. We follow the Theory of Didactical Situations within the Didactical Engineering methodology (both widely used in mathematics education research) to design and analyse a didactical situation about asymmetric cryptography using graphs. Following the phases of Didactical Engineering, after the preliminary analysis of the content, the constraints and conditions of the teaching context, we conceived and analysed the situation a priori, with a particular focus on the milieu (the set of elements students can interact with) and on the choices for the didactical variables. We discuss their impact on the problem-solving strategies the participants need to elaborate to decrypt an encrypted message. We implemented our situation and collected qualitative data. We then analysed a posteriori the different strategies that participants used. The comparison of the a posteriori analysis with the a priori analysis showed the learning potential of the activity. To elaborate on different problem-solving strategies, the participants need to explore and understand several concepts and methods from mathematics, informatics, and the frontier of the two disciplines, also moving between different semiotic registers.

Keywords: public-key cryptography, unplugged activity, pre-service teacher training, didactical engineering, theory of didactical situations, interdisciplinarity, perfect dominating set.

* Corresponding author.

1. Introduction

In the last decade, the importance of introducing informatics[1] in K-12 education has been strongly advocated (Wilson *et al.*, 2010; CECE (The Committee on European Computing Education), 2017). Informatics should be recognised as a fundamental, independent scientific discipline to be taught to students, so they can understand the digital world we are immersed in and become active and informed citizens and, potentially, workers in the ever-increasing digital job market.

However, in our increasingly complex and rapidly changing world, many criticise the traditional siloed teaching of disciplines in school and advocate a much more integrated, interdisciplinary teaching, particularly for the fundamental STEM (science, technology, engineering, and mathematics) fields (Millar, 2020). One difficulty in developing STEM activities at the secondary school level is the lack of training in this direction for teachers in the involved disciplines.

In the context of the IDENTITIES Erasmus+ European Project, a larger project about interdisciplinarity in STEM education and pre-service teacher training, we developed a teaching activity on public-key cryptography. The activity was designed for teacher training and was tested during pre-service teacher training events. Furthermore, although not yet tested in that context, its content and organisation have the potential to be used by teachers for classroom activities and/or projects with high school students.

The activity we developed has the objectives of teaching the big ideas and challenges of public-key cryptography and making participants interact with the interdisciplinary objects (pertaining to informatics and mathematics) the activity contains. We chose to design a public-key cryptography activity (based on a computationally hard problem on graphs) for epistemological and didactical reasons. Epistemologically, informatics and mathematics are deeply interconnected in the cryptography research field and discipline, and the activity, as we will see, can bring up many topics like algorithms, computational complexity, graphs, matrices, and linear systems. Educationally, informatics and cryptography are well suited to provide *adidacticity*, which is the potential to enable learning independently of teacher interventions (see Subsections 2.2 and 3.2).

For the content of our activity, we relied on a cryptosystem first described by Fellows and Koblitz (1994), based on the problem of finding a perfect dominating set in a random graph. Bell *et al.* (2003, pp. 209–211) developed a CS Unplugged activity for high-school students using that cryptosystem. We will use the same cryptosystem, although our design differs significantly from the original.

To design the teaching activity we followed the Theory of Didactical Situations (Brousseau and Warfield, 2020) within Didactical Engineering (Artigue, 1994), a research methodology widely used for decades in mathematics education research. The main objective is 'the controlled design and experimentation of teaching sequences [...] adopting an internal mode of validation based on the comparison between the a priori and a posteriori analyses of these' (Artigue, 2020).

[1] We use the term 'informatics' as a European synonym for computer science (CS) or computing.

Section 2 presents our theoretical and methodological underpinnings: Didactical Engineering (2.1) and the Theory of Didactical Situations (2.2).

The subsequent sections are organised according to the phases of the Didactical Engineering research methodology. Section 3 presents the preliminary analysis of the current epistemological, institutional, and didactical context of interdisciplinary and cryptography teaching. Section 4 details the chosen cryptosystem's computational, mathematical, and didactical aspects, which are necessary to understand our activity. Section 5 describes the research purposes of the study and the design of the situation. Section 6 details the a priori analysis of the didactical variables and their impact on the different problem-solving strategies and their relative interdisciplinary potential. Section 7 describes the implementation of our situation and the data collected. Section 8 presents our a posteriori analysis in light of the a priori one. Section 9 discusses our results and gives concluding remarks.

2. Methodology

2.1. *Didactical Engineering*

For this research, we chose the Didactical (or Didactic) Engineering (DE) methodology (Artigue, 1994, 2020), which consists in designing and analysing (from an epistemological and a didactical point of view) a situation that is experimented with afterwards. DE is a *qualitative* research methodology proposed by Guy Brousseau in the early eighties and successfully practised for research in didactics of mathematics for several decades. This methodology, which is centred on the conception, organisation, and study of classroom realisations, was developed by French researchers in order to address both theoretical and practical aspects of the science of didactics that could not be captured by the existing methodologies adapted from other scientific fields such as psychology (questionnaires, interviews, and test comparisons). DE aims to build a constructive relationship between research and practice: didactical systems are considered in their concrete functioning, and as such, the researchers should take into account the conditions and constraints under which the teaching and learning process is done. DE has also been practised outside the community of mathematics education (for example, in physical sciences education) and has been used for teacher education and to test new pedagogical techniques.

Starting from the content to be taught (in our case, public-key cryptography), the DE process is structured in four phases:

1. The *preliminary analysis*. This first phase clarifies the background for the next phase, i.e., for the conception and organisation of the didactical situation. It consists of studying the mathematical (and informatics, in this case) content and the conditions of the teaching and learning process. This analysis is done in three different dimensions.
    - An institutional analysis of the constraints and conditions where the DE methodology will take place. These conditions and constraints can be of

different natures: curricular characteristics, teaching practices, technological resources available, evaluation practices, characteristics of the students and of the teachers involved, and so on.
- An epistemological analysis of the content. In this part, possible epistemological issues that may be related to the content of the situation are identified.
- A didactical analysis which is mainly a survey of the existing research literature about teaching and learning the content of the situation.

2. The *conception and a priori analysis*. This second phase consists of modelling the situation and analysing its content and organisation within a theoretical didactical framework – in our case, the *Theory of Didactical Situations*[2] (explained later in 2.2). The conception and the analysis are closely related in the sense that the analysis helps revisit and adapt the conception in order to achieve the teaching and learning objectives.

   Conception requires a number of choices that may concern different levels of organisation and of disciplinary content. During the a priori analysis, these choices are made explicit, and their relation to the research hypothesis and to the preliminary analysis is clarified. These choices may concern the task itself, its content, but also the resources proposed to the students. The way these choices affect the possible strategies conducted by the students to solve the problem is discussed during the a priori analysis in order to reveal possible obstacles, interactions and dynamics. The goal of the a priori analysis is not to predict individual student behaviour but to create a generic reference of the learning potential of the situation designed and its potential difficulties. This reference will be used afterwards to compare with classroom realisations.
3. The *realisation, observation and data collection*. During the implementation, the researchers collect data to be used for the a posteriori analysis. The data collected aim to understand the interactions of students with the *milieu* (the set of elements with which the student can interact, see Subsection 2.2) and to understand at what point the choices made help students move from initial (naive) strategies to more elaborate and complex strategies that involve potential learning. Usually, the data collected are observers' notes, students' productions and files, and audio or video recordings. The researchers are in the position of observers during the realisation phase.
4. The *a posteriori analysis*. The a posteriori analysis concerns the comparison of the data collected during the classroom realisation with the a priori analysis. What were the convergences and divergences, and what do they reveal? What were the interactions that were not anticipated? How can we interpret the contrast with respect to the difficulties and the learning potential of the situation? Note that there will always be differences between the realisation and the a priori analysis because the a priori analysis considers an abstract generic student

[2] Other frameworks, like the *Anthropological Theory of the Didactic* (Chevallard and Bosch, 2020) are not discussed in this work.

behaviour which is, of course, never present during a real classroom implementation. As such, the validation of the research hypotheses does not require an exact match between the a priori and the a posteriori analysis. Nevertheless, the understanding of students' activity is made possible because of the depth of the a priori analysis.

In summary, according to DE methodology, the a priori analysis is compared with the a posteriori analysis of the experimentation. The validation of research hypotheses is *internal*, i.e., results from the epistemological conformity of a posteriori analysis with a priori analysis. Moreover, the a priori and a posteriori analyses develop in a circular process, where each experimentation can contribute to enriching and refining the a priori analysis.

2.2. *Theory of Didactical Situations*

For the development of this specific DE process, we rely on the Theory of Didactical Situations (Brousseau and Warfield, 2020).

For our analysis, we use several concepts from the Theory of Didactical Situations:

- A *didactical variable* is a variable of the problem or situation whose values influence the possibility or the hierarchy of strategies that students implement to solve the problem. Identifying the didactical variables and their effects and choosing their values according to the learning objectives is a crucial element of the a priori analysis.
- The *milieu*, in the sense of the Theory of Didactical Situations, is the set of situations' elements with which the student can interact. In response to students' actions, the milieu produces retroactions, allowing them to adjust their behaviour, modify their understanding of the problem, and adapt, i.e., learn. The analysis and organisation of a milieu, the possible actions, and the retroactions that the milieu allows is a central element of a didactical situation and its a priori analysis.
- The *adidacticity* of a situation is the potential of a situation and its milieu to enable learning independently of teacher interventions. In the Theory of Didactical Situations, adidacticity in learning is essential, mainly because it prevents specific side effects of the didactical contract (the implicit pact between students and teachers (Brousseau *et al.*, 2020)).
- For this learning potential to be realised, the teacher must transfer part of the responsibility for solving the problem to the students. The *devolution* allows this transfer of responsibility (Brousseau and Warfield, 2020) and must be taken into account when designing a didactical situation. Conversely, *institutionalisation* is the teacher's activity that allows students to structure into more formal knowledge what they have learned from solving the problem.

We rely on these concepts to design, organise and analyse the didactical situation presented in what follows.

3. Preliminary Analysis

3.1. *Institutional Analysis (Elements of Context)*

The didactical situation reported in this paper is part of a module on cryptography for prospective science teachers (who are the target learners for the module). The module is one of the outputs of the IDENTITIES European project involving five universities that aims to design novel teaching approaches to interdisciplinarity in science to innovate pre-service teacher education. The project developed and tested innovative teaching modules on interdisciplinary curricular topics (such as cryptography) to explore inter-multi-transdisciplinary knowledge organisations and to develop interdisciplinary classroom activities and new models of co-teaching. The teaching modules aim to highlight and question the identities of STEM disciplines through reflections on their epistemological and linguistic structures, focusing on the interaction between physics, mathematics, and informatics.

Each module must last about 6 hours. Its interdisciplinary content must be socially relevant and potentially suitable for high school students as well. Indeed, the content must be understandable to high school teachers of STEM disciplines without discipline-specific prerequisites. Module activities must be easy to understand for all the participants yet engaging and approachable in the given time.

These interdisciplinary modules were implemented and tested twice in week-long training schools for student teachers. The first training school took place in 2021 and was held online because of the COVID-19 pandemic restrictions. This first remote implementation of our cryptography module informed our design and helped us both refine the teaching activity and design an observation grid for researchers to use during the implementation. In this paper, we will focus on the analysis of the teaching activity and its implementation during the second training school, which took place in 2022 in physical presence. Twenty-eight student teachers participated, five or six from each partner's institution. They had a disciplinary background (bachelor's or master's degree) in informatics, mathematics, physics, or natural sciences, and experience or motivation in science education[3]. The training school was held in English (which was not the native language of any of the participants). Each module was attended by about 14 prospective teachers.

Therefore, the institutional constraints for the design of our didactical situation were the following. (i) Three hours (out of six) available for the didactical situation during the module. (ii) Fourteen participating student teachers with different disciplinary and linguistic backgrounds. (iii) No specific disciplinary prerequisites of the participants could be assumed. (iv) Participants could use PCs and tablets, and Internet access was provided.

[3] The participants had to be enrolled in a master's program (or equivalent course, depending on the national regulations) to become high-school teachers in STEM disciplines.

3.2. *Epistemological and Didactical Analysis*

The incredibly rapid development of our digital society has widened the gap between what is taught in schools and what students experience in daily life. One of the most significant causes may be the rigid discipline-based organisation of the school curriculum (Miani, 2021). STEM movement tries to answer this challenge by proposing the integration of science, technology, engineering, and mathematics in an interdisciplinary and applied approach that deals with real-world problems and problem-based learning. According to the movement, these subjects do not exist in isolation in the real world, and therefore they should not be taught separately (STEM Task Force, 2014, p. 11). However, the 'traditional siloed subject teaching of STEM' is far from being overcome. Many challenges are still to be tackled, like 'inadequate teacher knowledge incorporating all STEM fields, and the lack of materials and instructional and assessment support and guidance'. Moreover, teachers 'struggle to make connections across the STEM disciplines [...and] expressed difficulty in using frameworks from other disciplines[...] and felt [...not] able to impart meaningful learning' (Miani, 2021). Also, it is still fundamental to value the specific characteristics, methods, and ways of thinking of the different disciplines in order for a fruitful interdisciplinary interaction (Barelli *et al.*, 2022).

In this paper, we focus on the interdisciplinarity between informatics and mathematics. The disciplines have 'strong links and a common history', sharing common foundations, 'fields developing at their interface', and 'a very similar relation to other sciences through modelling and simulation' (Modeste, 2016, pp. 243–244). Cryptography is one of the fields that is developing at the interface of informatics and mathematics (Modeste, 2016), and therefore it is a good candidate for our activity.

The Cybersecurity Curricula 2017 (Joint Task Force on Cybersecurity Education, 2018) gathers guidelines for graduate programs in cybersecurity and indicates cryptography as necessary to lay the foundation for subsequent learning. Suggested content includes basic cryptography concepts, the necessary mathematical background, and symmetric and asymmetric cyphers; recommended methodologies include concrete encryption and decryption activities. Similar indications are also found in pre-university (K-12) informatics education standards from CSTA (CSTA, 2017), which also recommend unplugged hands-on activities. However, a review of ACM education conferences between 2010 and 2019 (Švábenský *et al.*, 2020) shows that most publications on informatics education investigate cybersecurity learning, where cryptography is seen as only one of many topics (e.g., Sommers, 2010; Turner *et al.*, 2011; Brown *et al.*, 2012; Deshpande *et al.*, 2019) and often viewed only from a technical and instrumental perspective rather than for its fundamental principles and social implications. Nevertheless, some significant indications emerge from a review of the works that deal specifically with cryptography in school settings. First, hands-on, cooperative, and inquiry-based activities can improve students' self-efficacy and problem-solving skills (Konak, 2018). In addition, the use of didactical tools (e.g., Simms and Chi, 2011; Schweitzer and Brown, 2009; Ma *et al.*, 2016; Anane and Alshammari, 2020) to

visualise and simulate how cryptosystems work, their weaknesses, and possible attacks, is recurrent. However, such tools are often too rich in technical details for novices and nonspecialist students; furthermore, their interactivity is limited to changing some parameters of the simulations. Some authors have proposed and implemented unplugged activities in which students experience encryption and decryption algorithms, protocols, and attacks at a high level and without computers (e.g., Bell *et al.*, 2003; Konak, 2014; Fees *et al.*, 2018; Rosamond, 2018). These activities use simple objects (e.g., scissors, maps, padlocks) and concrete actions (e.g., cutting out paper, mixing colours). Rosamond (2018) describes two unplugged cryptography enrichment activities: one based on covering a directed graph with vertex disjoint cycles and one based on perfect codes or perfect dominating sets. As anticipated, the latter was introduced by Bell *et al.* (2003): in the activity, a one-way function is simulated by constructing a graph with a perfect dominating set (i.e., a particular subset of the graph nodes). Such a graph is the base for a cryptosystem that uses elementary arithmetic computations to encrypt a number. This is the basis of our didactical situation and is explained in detail later as such (see Section 4).

Communicating in secret and trying to decrypt messages without knowing the key is engaging and motivating for students (Lindmeier and Mühling, 2020). Moreover, cryptanalysis has an inherent potential for *adidacticity*, that is, the potential for learning with a strong autonomy left to students' interactions with the problem. Indeed, if one is trying to find the secret key of a cryptosystem (where encryption and decryption algorithms are public), the supposed key can be tested by decrypting the messages that have been encrypted with that key and see if the result is the original plaintext message[4]. Considering these two aspects of cryptography, we have organised a didactical situation based on a public-key cryptosystem. Based on bibliographic research and the analysis of several proposals, we chose a cryptographic system based on a graph theory problem: the existence of a perfect dominating set on a graph. The choice was guided by the need for the activity to be understandable by students with no informatics or cryptography background and to involve interdisciplinary objects like graphs.

In the following, we give some definitions and formalise the cryptosystem.

4. A Public-key Cryptosystem Using Perfect Dominating Sets on Graphs

In an encryption scheme, we assume two individuals communicate on a public channel. In order to ensure the confidentiality of their communication, the parties use an *encryption algorithm* to transform a *plaintext* message into an *encrypted* message (a ciphertext). The security of this process is based on (one or several) *keys*, allowing both parties to encrypt and decrypt messages. There are two types of cryptosystems: symmetric (or secret-key) and asymmetric (or public-key). In a symmetric cryptosystem, the encryp-

[4] Analogously, in computer programming, students can autonomously test their program and check if it works by comparing the desired and actual result of the computation without waiting for teacher's validation.

tion and decryption key is the same. In an asymmetric cryptosystem, the encryption key (public key) and the decryption key (private key) are different.

In our work, we will focus on public-key cryptosystems. The main elements of a public-key encryption scheme[5] are: a key generation algorithm Gen that generates a pair of keys (pk, sk), i.e., a *public key* and a *private* (or *secret*) *key* for each user; an encryption algorithm Enc that, given the pk of the receiver and a plaintext message m, outputs a ciphertext message $c = \mathsf{Enc}_{pk}(m)$; a decryption algorithm Dec that, given the receiver's private key sk and a ciphertext c, outputs a plaintext $m = \mathsf{Dec}_{sk}(c)$ identical to the plaintext encrypted with the public key pk. Both functions Enc and Dec should be easy (that is, efficient) to compute if the keys pk and sk, respectively, are available. The security of the scheme depends on the difficulty (that is, the computational complexity) of computing the function Dec *without* knowing the private key sk.

Fellows and Koblitz (1994) proposed an asymmetric cryptosystem based on a difficult problem: the Perfect Dominating Set (PDS) problem.

Let a graph $G = (V, E)$ with V the set of vertices and E the set of edges. A (closed) neighbourhood of a vertex $u \in V$ is the set $N[u] = \{v \in V \mid uv \in E\} \cup \{u\}$, of vertices of V adjacent to u as well as u (in other words, all vertices of distance $\leqslant 1$ from u). A *dominating set* of G is a subset of vertices $S \subseteq V$ such that every vertex of V is included in the neighbourhood of a vertex of S. If S is a dominating set of $G = (V, E)$, then every vertex of V is a neighbour to at least one vertex of S, or it belongs to S. If each vertex of V is included in exactly one neighbourhood of a vertex of S, then S is said to be a *perfect code*, often referred to also as *perfect dominating set* (noted PDS in the following). Fig. 1 gives an example of a graph with a PDS.

A useful result is that if a graph has more than one PDS, they all have the same size (Klostermeyer, 2015, p.106).

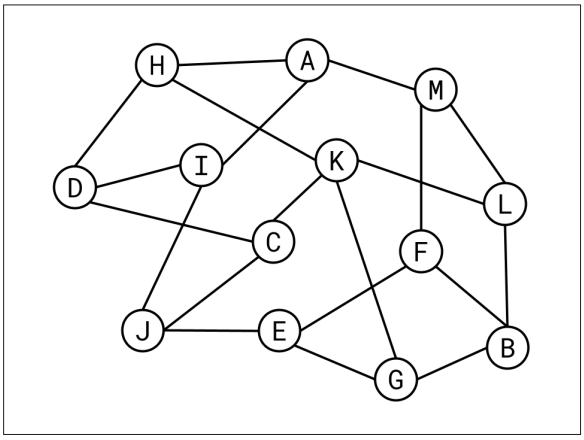

Fig. 1. $\{I, K, F\}$ is a PDS of this graph.

[5] For a formal definition, see for example Katz and Lindell (2007, p. 366)

Thus, the PDS problem is the following (Fellows and Hoover, 1991; Haynes *et al.*, 2013):

PDS Problem	
Input:	A graph $G = (V, E)$
Output:	A PDS of G (if one exists)

In general, deciding whether there exists a PDS in a given graph is an NP-complete decision problem (Klostermeyer, 2015, p. 107), and therefore finding a PDS in a given graph (our PDS Problem) is a NP-hard problem[6].

This means that we only know algorithms that take exponential time with respect to the number of nodes, and we don't know if we will ever be able to do better than that. As we will explain, we can use this feature to design a cryptosystem.

For our didactical situation, we used an instance of the PDS problem, i.e., we have constructed a graph with a PDS. The choice of this graph is important, as will be explained later.

Using the PDS problem, we can design a cryptosystem based on the following two facts:

- Given a set of vertices, we can easily construct a graph whose PDS will be this set of vertices.
- Given a graph containing a PDS, it is difficult to find the PDS if we only know the graph.

The PDS cryptosystem is the following: Alice and Bob want to communicate confidentially. Bob wants to send a message m (in this case, m is an integer) to Alice. They use the following encryption protocol:

1. Alice builds a graph $G = (V, E)$ with a PDS S. The graph G is Alice's public key, and the PDS S is Alice's private key. Let $V = \{v_1, v_2, ...v_k\}$.

[6] We summarise here, in an informal way, the most relevant ideas. NP-completeness is a vast topic in the study of the computational complexity of problems: for a formal introduction, we suggest Cormen *et al.* (2022, ch. 34).

We say a problem is in the set P if we can solve it in polynomial time with respect to the input size (i.e., it can be solved in $O(n^k)$ time for some constant k, with n the input size).

We say a problem is in NP if we can verify a solution (or, more formally, a solution 'certificate') in polynomial time with respect to the input size. Intuitively, $\mathsf{P} \subseteq \mathsf{NP}$, but whether $\mathsf{P} = \mathsf{NP}$ or $\mathsf{P} \neq \mathsf{NP}$ is one of the most famous, relevant and open questions of Informatics.

We focus on a particular set of NP problems: the NP-complete problems. These problems are considered 'the most difficult NP problems' because if we find a polynomial-time solution for one of them, then we can solve all the NP problems in polynomial time. NP-complete problems include relevant problems for today's world. Still, unfortunately, no one has ever found a polynomial solution for any of them, nor has it been proven that such a polynomial solution cannot exist.

Formally, the NP-complete set includes only *decision* problems, i.e. those problems whose output is either a 'yes' or a 'no'. For example, as said, determining whether a graph has a PDS is an NP-complete problem. Since we are interested here in the complexity of finding an actual instance of that PDS, we are not dealing with a decision problem. Still, it should be easy to convince yourself that our problem is *at least as hard as the decision problem*. Therefore, we say that finding a PDS on a given graph (our PDS problem) is NP-hard.

2. Bob chooses integers m_1, m_2, ..., m_k such that $m_1 + m_2 + ... + m_k = m$.
3. Bob writes on each vertex v_i of V an m_i. We call m_i the *secret value* of the vertex v_i.
4. For each vertex v_i, Bob sums its secret value with the secret values of its neighbours. This new value p_i is called the *public value* of the vertex v_i.
5. Bob writes on each vertex its public value and erases the secret values. The encrypted message is the graph G with the public values.

Fig. 2 gives an example of a graph with its public and secret values.

To decrypt the message, Alice computes the sum of the values on the vertices of the PDS (Alice knows the PDS because it is her private key). Note that the graph G (public key) and the encrypted message (graph G with public values) can circulate without an eavesdropper being able to read the plaintext message (a priori). Note also that if the graph has several PDS, then any PDS can be used for decryption[7].

The security of the system is based on the fact that it is (NP-)hard to find the PDS given the graph. Of course, the PDS cryptosystem is only '*didactically* secure' because simple algebraic attacks are possible (Fellows and Koblitz, 1994), as we will see.

As said, the PDS cryptosystem was first presented by Fellows and Koblitz (1994), and an unplugged activity based on it has been included in the Classic CS Unplugged (Bell *et al.*, 2003, 2015).

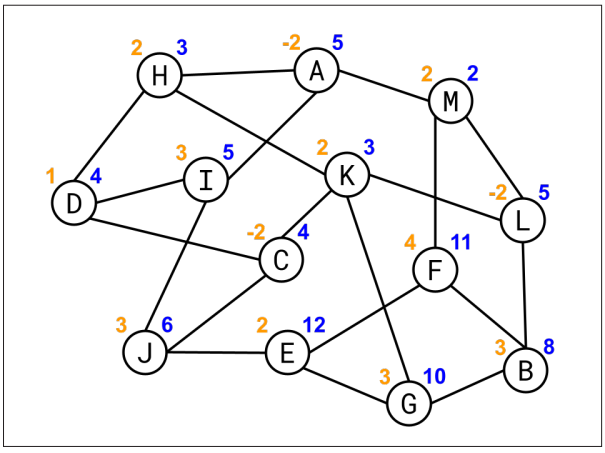

Fig. 2. Example of an encrypted message using a graph G. Secret values in orange (left of the node) and public values in blue (right of the node). The plaintext message m is the sum of the secret values ($m = 19$).

[7] Note that proving that decryption is correct does not use the uniqueness of the PDS. The public value of a node v in (any) PDS is the sum of the secret values of the 'stars' (see Fig. 3) whose centre is v. Since every node of the graph is connected to exactly one node of a PDS by definition, summing up the public values of all the nodes of a PDS will result exactly in the sum of all the secret values.

5. Conception

5.1. *Research Purposes*

As explained above, this research was developed inside a European project whose more general goal is to create innovative teaching modules for pre-service teacher training about interdisciplinarity in STEM fields (with a particular focus on links and interactions between physics, mathematics, and informatics).

In that context, we designed, implemented, and analysed a didactical situation for pre-service STEM teachers to introduce the idea of public-key cryptography and make them explore fundamental informatics and mathematical concepts and methods.

For this work, the research purposes are, therefore, the following:

RP1 Examine the different strategies (analysed both a priori and a posteriori, after an actual implementation) that student teachers will adopt to decrypt a message starting from different information at their disposal (*access to information* is the main didactical variable of the situation, see Subsection 6.2).

RP2 Examine how student teachers will interact with different disciplinary and interdisciplinary objects like matrices and graphs, using methods and practices from mathematics and informatics, moving between different semiotic representations[8].

As said, the cryptosystem is already known (Fellows and Koblitz, 1994) and used for didactic purposes (Bell *et al.*, 2003, 2015). However, the novelty of our approach was to embed this problem in a didactical situation (in Brousseau's meaning), developed and implemented within the methodological framework of didactic engineering, aiming at fostering the interdisciplinarity of mathematics and informatics. Our contribution is the precise organisation of its milieu and the analysis of the didactical variables that come into play, together with specific choices for their values. Moreover, we propose an implementation in teacher training that takes into account the strengths of the PDS problem.

During the didactical situation, the participants are given a problem (to decipher a message) that is not broken down into simpler tasks. Because of this fact (and because of the careful choices of the didactical variables), the participants need to elaborate specific strategies in order to address the problem. These strategies (explained in the next section) require the use and understanding of several concepts and methods from mathematics and informatics and sometimes the change of semiotic registers.

In what follows, we describe the choices of the didactical variables and the resulting strategies, underlining the concepts and the methods involved.

The more general research purpose of examining this activity's learning potential and impact on related disciplinary and interdisciplinary concepts is left for future work.

[8] Intuitively, the *theory of registers of semiotic representation* (Duval, 1995, 2017) is based on the fact that 'there are as many different semiotic representations of the same mathematical object as semiotic registers utilised' (Pino-Fan *et al.*, 2015).

5.2. *The Didactical Situation*

The objectives of the didactical situation are the following:

- Introduce some general concepts and terminology of cryptography (plaintext and encrypted message, encryption and decryption algorithms, key, attack models, private and public keys, difficult-to-reverse problem, one-way function, and so on) and make students understand and explore the principles and issues of public-key cryptography.
- Make students explore and interact with mathematical and informatics concepts and objects on the boundary of the two disciplines (such as graphs, algorithms, and matrices).

The didactical situation is organised as follows:

Step 1: Encryption. We explain the encryption algorithm to the participants by means of a graph G (the public key). We do not introduce or explain the notion of PDS (it is not needed to encrypt a message). We do not say that G has a PDS either.

Step 2: Cryptanalysis. The participants are divided into three groups. All the groups are given the same encrypted message (i.e., graph G with public values on it) and asked to decrypt it. Each group is given different information to solve the problem.

- Group A is given the definition of PDS and the (unique) PDS for the given graph G. We do not explain the decryption algorithm. Group A is in the position of a cryptographic engineer who has all the mathematical elements available and needs to combine them to design a public-key cryptosystem.
- Group B is given the definition of PDS and the decryption algorithm (which uses the PDS). They do not know the PDS for graph G. Group B is in the position of a cryptanalyst carrying out a *person-in-the-middle attack*; that is, the attacker has knowledge of the public-key cryptosystem but does not know the private key.
- Group C has no information other than the encrypted message itself. Group C does not know the decryption algorithm. There is also no reference to the existence of a PDS. Group C is in the position of a cryptanalyst trying to find the plaintext message without necessarily finding the private key.

All groups can independently check whether they have decrypted the message correctly.

6. A Priori Analysis

6.1. *A Priori Analysis Elements*

In the following, we schematically describe strategies that the groups can use to decrypt the message given their available information. Note that students work on a

graph that is larger than the previous examples, as discussed in Subsection 6.2 and shown in Fig. 4.

6.1.1. *Group A*

Available Information. The definition of PDS and the (unique) PDS for the given graph G. Note that group A is not given any elements on how to use the PDS to decrypt: their goal is to find by themselves the decryption algorithm using the PDS.

Strategy. Identify the neighbourhoods of all vertices that belong to the given PDS. Then observe that the intersection of these neighbourhoods is empty and that the union of these neighbourhoods covers the graph G. The neighbourhoods can be represented as lists of vertices or graphically as 'stars' on the graph (see Fig. 3). By construction of the cryptosystem, the public value of each vertex is the sum of the secret values of its neighbourhood. Thus, the sum of the public values of the vertices of the PDS is equal to the sum of the secret values of all the nodes, which is the plaintext message.

The definition of PDS is expressed using terminology from set theory. In order to elaborate this strategy, group A needs to interpret this definition on the graphical representation of the graph and make the connection with the encryption procedure. More precisely, they need to deduct what the perfect domination property means for the public values of the nodes. This procedure is not trivial and requires an intuitive understanding of the proof of correctness of the cryptosystem. This way, group A has the potential to explore the central idea of such proof, i.e., the decryption of an encrypted message returns the plaintext message $\mathsf{Dec}_{sk}(\mathsf{Enc}_{pk}(m)) = m$. This can be the object of a formulation phase consisting of making the decryption algorithm explicit and of a validation phase of proving the encryption's correctness.

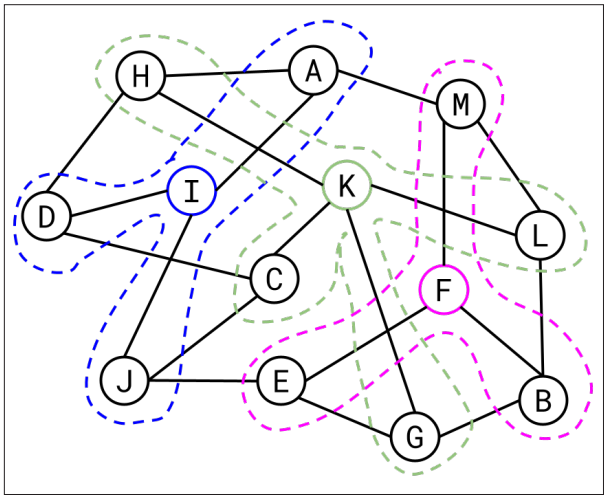

Fig. 3. We can visualise the 'stars' with the PDS nodes as centres, showing that each node is directly connected to exactly one node of the PDS.

6.1.2. *Group B*

Available information. The definition of PDS and the decryption algorithm (which uses the PDS). Group B knows that there is a PDS in graph G, but they do not know which nodes are the PDS on that specific graph. This incites the group to try to find the private key (the PDS) using the encrypted message and the public key (G with the public values noted on its vertices). Group B is thus confronted with an instance of the difficult problem of finding a PDS in a graph.

Strategies. We describe three possible general strategies for this group. These strategies are interesting because they use different semiotic registers (Duval, 1995, 2017): the graph representation, the lists of vertices, and the adjacency matrix of the graph. These three strategies amount to a structured, exhaustive search of the subsets of vertices to find a PDS that is known to exist. This search can be done in an organised and structured way (algorithm) or in a more heuristic way, based on the same principles.

Strategy 1: Finding stars in the graph. Let a graph $G = (V, E)$ and let S be a PDS of G. This strategy is based on the following ideas:

- Let v be a vertex of V. By the definition of the PDS, in the neighbourhood $N[v]$, there exists exactly one vertex that belongs to S. Thus, if v vertex is not in S, then exactly one of its neighbours is in S.
- If a vertex u belongs to S, then (a) the neighbouring vertices of u do not belong to S, and (b) for any neighbour u' of u, the neighbouring vertices of u' do not belong to S either (otherwise u' would be linked to two vertices that belong to S). Thus, if we find a vertex of S, we can deduce that its neighbours and the neighbours of its neighbours are not in S.

Informally, we add step-by-step vertices in a set S in order to find a PDS. When we do not succeed, we backtrack to the choices of vertices made to continue the exploration of potential PDS. In informatics, backtracking is a 'systematic way to run through all the possible configurations of a search space'. It is relevant especially when 'we must generate each possible configuration exactly once' (Skiena, 2020, p. 281). Intuitively, we build a solution incrementally, and when we reach a partial solution that cannot become a correct solution anymore, we abandon the path and backtrack to explore other paths.

Elaborating this strategy first requires understanding the definition of a PDS (which is expressed symbolically in set theory language) and then interpreting the definition on the graphical representation of the graph (by drawing subgraphs as stars, see Fig. 3). Systematising the steps of the algorithm requires an intuitive understanding of both the properties of domination and perfect domination and the idea of backtracking.

In Algorithm 1, we provide a more formal description of this strategy. Note that, although more rigorous, it is still informal in some operations.

In a heuristic approach to the above strategy, we start with a vertex v of a small degree to deal with a small number of starting cases.

Strategy 2: Lists. Let G be a graph and S a PDS of G. For each vertex of G, we write its neighbourhood as a list. We then study these lists in order to find a set of lists

Algorithm 1 Finding stars in the graph

$S \leftarrow \{\}$
Choose a vertex v, with neighbourhood $N[v]$
▷ We are sure that v or one of its neighbours is in the PDS
while True **do**
    Choose $t \in N[v]$
    Add t to S
    repeat
        Mark red all nodes in $N[t]$
        ▷ A previous green may be overridden with red, if necessary
        Mark green any non-red neighbour of neighbours of t
        ▷ i.e. the non red nodes in $N[x]$ for all $x \in N[t]$
        if there is an uncoloured vertex w connected to a green vertex **then**
        ▷ Backtracking point
            Choose such w and add it to S
            $t \leftarrow w$
            Done ← False
        else
            Done ← True
        end if
    until Done **or** S is aPDS
    if S is a PDS **then**
        return S
    else if backtracking is possible **then**
        ▷ i.e. if we may choose some other vertex at one backtracking point
        Backtrack (undoing the colouring and the additions to S) to the last possible backtracking point
        Choose a different w and start again from there
    else
        Remove all the colouring ▷ We want to iterate again with a new t
        Remove t from $N[v]$
        $S \leftarrow \{\}$
    end if
end while

whose intersection is empty and whose union covers the graph G. The basic idea of this strategy is that each vertex of the graph G belongs to exactly one neighbourhood of a vertex of S.

Informally, the idea is to incrementally build a collection L of lists ℓ_i, such that the intersection of the $\ell_i \in L$ is empty, while their union contains all the vertices of G.

More rigorously, we formalised this strategy in Algorithm 2, where L is a LIFO stack since the first element removed is always the last one added. In informatics, a *stack* is a collection of elements that implements the LIFO (last-in, first-out) policy:

Algorithm 2 Merging lists

$V \leftarrow$ the set of vertices of G ▷ Vertices are enumerated starting from 1
for all $v_i \in V$ **do**
    $\ell_i \leftarrow N[v_i]$ as a list
end for
$L \leftarrow$ *emptystack* ▷ L is a LIFO stack; any item in L is a list ℓ_i
$k \leftarrow 0$
while $\bigcup L \neq V$ **do** ▷ $\bigcup L$ is the union of all the lists ℓ_i in L
    if there exists $i > k$ such that $(\bigcup L) \cap \ell_i = \emptyset$ **then**
        Let i be the min index that satisfies the condition
        Push ℓ_i onto L
        $k \leftarrow i$
    else
        Pop (remove) the top item g_j from L
        $k \leftarrow j$
    end if
end while
return the set $\{v_i : \ell_i \in L\}$ (that is a PDS)

like in a pile of plates, you can only *push* a new plate on top of the stack, or *pop* the plate on the top of the pile. Therefore, the 'order in which plates are popped from the stack is the reverse of the order in which they were pushed onto the stack, since only the top plate is accessible' (Cormen *et al.*, 2022, p. 254): the last plate you pushed in is the first you pop out.

Elaborating this strategy requires understanding the domination and perfect domination properties, expressing these properties using lists, and also an intuitive understanding of a LIFO stack (even if it is not necessarily recognised as such).

Strategy 3: Adjacency matrix of the graph. This strategy consists in writing the adjacency matrix of the graph G and in selecting a set of rows whose sum is a row of 1. Indeed, in the adjacency matrix, for a vertex i, in the corresponding row $l_i = [a_{i1}, a_{i2}..., a_{in}]$ the coefficients $a_{ij} = 1$ if the vertices j and i are connected and 0 otherwise. Note that here $a_{ii} = 1$ for every vertex i (because, in the PDS definition, we are considering closed neighbourhoods). Thus, if we find a set of rows whose sum equals [1, 1, ..., 1], the vertices corresponding to these lines constitute a PDS (because each vertex of G is adjacent to exactly one of the chosen vertices).

The idea of strategy 3 is very close to that of strategy 2. Still, the register of representation is different: on the same scheme as algorithm 2, we go through the set of rows of the matrix, including or excluding rows, to find a subset of rows whose sum equals [1, 1, ..., 1].

Elaborating this strategy requires, once again, understanding the properties of the PDS definition and expressing these properties using the adjacency matrix.

6.1.3. *Group C*

Available information. No information other than the encrypted message. Group C only knows the encryption algorithm and does not know that a PDS exists in the graph nor how it can be used to decrypt. This group is asked (implicitly) to search for possible flaws in the cryptosystem without necessarily searching for the private key.

Strategy. Starting from the encrypted message, we form a linear system as follows: for each vertex v, of public value p_v and neighbourhood $N[v] = [v, v_1, ...v_k]$, we write the equation $x_v + x_{v_1} + ... + x_{v_k} = p_v$ where x_i is the secret value of vertex i. This equation translates the encryption step that allowed passing from private values to public values. We thus build a system of linear equations with as many equations and unknowns as there are vertices in G. The solution of the linear system is the tuple of all secret values $[x_1, x_2, ..., x_n]$, whose sum is the plaintext message m.

The linear system can be formed using the adjacency matrix of the graph G or by writing the linear equations for each vertex. We highlight that, in this activity, there is a correspondence between the adjacency matrix (one of the standard ways to represent the graph data structure in informatics (Cormen *et al.*, 2022, p. 549)) and the matrix equation that can be used to solve the linear system associated with the encrypted message on the graph. For example, the graph in Fig. 1 can be represented by the following adjacency matrix (note that, as said, the diagonal is all 1s because, even if edges from each node to itself are not drawn, each node is a neighbour of itself in the PDS definition)

	A	B	C	D	E	F	G	H	I	J	K	L	M
A	1	0	0	0	0	0	0	1	1	0	0	0	1
B	0	1	0	0	0	1	1	0	0	0	0	1	0
C	0	0	1	1	0	0	0	0	0	1	1	0	0
D	0	0	1	1	0	0	0	1	1	0	0	0	0
E	0	0	0	0	1	1	1	0	0	1	0	0	0
F	0	1	0	0	1	1	0	0	0	0	0	0	1
G	0	1	0	0	1	0	1	0	0	0	1	0	0
H	1	0	0	1	0	0	0	1	0	0	1	0	0
I	1	0	0	1	0	0	0	0	1	1	0	0	0
J	0	0	1	0	1	0	0	0	1	1	0	0	0
K	0	0	1	0	0	0	1	1	0	0	1	1	0
L	0	1	0	0	0	0	0	0	0	0	1	1	1
M	1	0	0	0	0	1	0	0	0	0	0	1	1

which is exactly the matrix A in the matrix equation $A\mathsf{x} = \mathsf{b}$ (that represents the linear system of equations that can be used to find the secret values given the public values on the graph) where

$$A = \begin{pmatrix} 1&0&0&0&0&0&0&1&1&0&0&0&1\\ 0&1&0&0&0&1&1&0&0&0&0&1&0\\ 0&0&1&1&0&0&0&0&0&1&1&0&0\\ 0&0&1&1&0&0&0&1&1&0&0&0&0\\ 0&0&0&0&1&1&1&0&0&1&0&0&0\\ 0&1&0&0&1&1&0&0&0&0&0&0&1\\ 0&1&0&0&1&0&1&0&0&0&1&0&0\\ 1&0&0&1&0&0&0&1&0&0&1&0&0\\ 1&0&0&1&0&0&0&0&1&1&0&0&0\\ 0&0&1&0&1&0&0&0&1&1&0&0&0\\ 0&0&1&0&0&0&1&1&0&0&1&1&0\\ 0&1&0&0&0&0&0&0&0&0&1&1&1\\ 1&0&0&0&0&1&0&0&0&0&0&1&1 \end{pmatrix} \quad \mathbf{x} = \begin{pmatrix} A\\B\\C\\D\\E\\F\\G\\H\\I\\J\\K\\L\\M \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} 5\\8\\4\\4\\12\\11\\10\\3\\5\\6\\3\\5\\2 \end{pmatrix}$$

Groups A and B can also be tempted to use linear systems too (even if, for group A, this means not using the private key, which is available).

Unfolding this strategy requires interpreting the cryptosystem as a linear system and examining its solution. Note that the solution to the problem does not necessitate the solution of the linear system but just finding the sum of all secret values; this can be done by finding the rows corresponding to the PDS nodes (for example, using the third strategy of group B). The concepts that come into play when elaborating this strategy are matrices, linear systems and their solution, and also the correspondence of the adjacency matrix with the system's matrix.

6.2. *Didactical Variables*

In this section, we identify the didactical variables of the situation and the potential effects of their values on the solving strategies. This will allow us to select relevant values in relation to our learning objectives for teachers and their presumed prior knowledge. These variables have been identified through the study of the problem itself and pre-experimented with volunteer students.

Access to information. In our didactical situation, the main didactical variable is the access to information which differs for the three groups. We have analysed in the previous section the possible strategies that correspond to different values of this variable.

The type of the graph G. It should be hard to find the PDS in the graph G. So one should exclude certain types of graphs for which it is known that the PDS problem is not hard. For example, if the graph is a tree, there exists a fast algorithm that solves the PDS problem (Klostermeyer, 2015, p. 107). The PDS problem is hard for planar graphs, but we observed that the use of non-planar graphs makes the problem visually more difficult for the participants. Therefore, we decided to use a non-planar graph.

The size of the graph $\|V\|$. The graph must satisfy certain criteria that make it usable when dealing with humans. More precisely, it must be large enough so that an exhaustive search of the PDS will be hard or tedious. At the same time, it must be small enough so that writing the linear system generated would still be possible for participants.

The maximum degree of the graph and the difference of degrees between vertices. A large difference of degrees between the vertices of the graph potentially influences the starting point and the execution of Algorithm 1; the participants tend to consider that

the vertices that have a degree 'too low' or 'too high' have special properties and they usually start the algorithm from those nodes. In order to avoid this effect, it is desirable to use a graph that is 'almost' regular. Note that if the graph is regular (i.e., all nodes have the same degree k), there is an easy solution to get the plaintext message that does not require finding the PDS: in fact, if we add all the equations of the linear system and divide by $k + 1$, we get the plaintext message (because each m_i will be added exactly $k + 1$ times in encryption).

Moreover, note that the size of the PDS $\|S\|$ is always in the interval $\frac{\|V\|}{\Delta+1} \leqslant \|S\| \leqslant \frac{\|V\|}{2}$ where V is the set of vertices of the graph G and Δ is the maximum degree of the vertices of G. For a given number of vertices, if the size of the PDS is close to the minimum value, the vertices that belong to the PDS have more neighbours. For our experimentation of the didactical situation, we have chosen a graph with 22 vertices and with $\|S\| = 4$.

The plaintext message and its composition. The plaintext message is a positive integer number. This number is subsequently decomposed into the values m_i (secret values) such that $\Sigma_{i=1}^{n} m_i$ with $\|V\| = n$ the size of the graph. Then $m_i \in \mathbb{Z}$ and can be repeated. We have chosen a decomposition in m_i where the absolute value for all m_i is small not to add cognitive difficulty for the participants. In our case, a number between 20 and 100 seemed a reasonable choice.

Using (or not) a computer algebra system. We chose to give the possibility of using a computer algebra system to solve the linear system. Its use was optional and was only proposed if the participants had independently come up with the idea of writing the linear system.

6.3. *Learning Potential*

Following the strategies presented in this section, students have to (i) translate the PDS properties of the graph into properties involving lists, matrices, and the graph visual representation; (ii) manually do an exhaustive structured search in the matrix or list space or the graph's representation understanding intuitively the idea of LIFO stacks and backtracking; and (iii) understand and justify why their strategies are correct. When dealing with a linear system, they may try to reduce it, determine if there is a unique solution or many, and reflect on the complexity of solving linear systems. Because of the retroactions with the milieu, students have to mobilise concepts, methods, and practices from mathematics and informatics to overcome the obstacles they encounter, moving between semiotic representations (Duval, 1995, 2017) of interdisciplinary *boundary objects*[9], such as matrices and graphs. We believe that this is a step towards understanding the challenges of public-key cryptography and of the interdisciplinary objects involved.

[9] artifacts [that] can fulfil a specific function in bridging intersecting practices' (Akkerman and Bakker, 2011, p. 134)

7. Realisation, Observation, and Data Collection

In 2021, we piloted an early implementation of the cryptography module entirely online via video conference, which particularly impacted the modes of interaction. In particular, group activity suffered. It was hampered by the less natural interactions (further complicated by the use of English, which was not the native language of any of the participants) and the inability to physically work together on the graphs (even though they were available online, in collaborative editors). However, this preliminary implementation helped us improve the didactical situation. Indeed, we deepened the a priori analysis, developed a design more consistent with the preliminary analysis, and better defined the practical organisation (timing, mode, materials).

In what follows, we present the implementation of the cryptography didactical situation, exactly as described in Section 5, which took place as part of the 2022 school for pre-service teachers, this time held with the presence of the participants (see Subsection 3.1).

The experimentation took place during a one-day (6 hours) session that included a preliminary presentation on symmetric and asymmetric cryptography (and its use and relevance in our society), the didactical situation (3 hours), and a collective reflection on the interdisciplinary aspects that emerged from the groups' work, led by the instructors. The teaching material for the entire module, including the situation, is available at `https://identitiesproject.eu/cryptography/`. Here we focus on the didactical situation itself.

The didactical situation (about 3 hours) constitutes its active learning part. The core of the didactical situation is an autonomous group activity (about 1 hour): a decryption challenge on which each group is then asked to report back to the other groups.

The participating student teachers were organised into three groups (A, B, and C) needed for implementing the didactical situation, as described in 6.1. The groups of 4 or 5 were composed so that the members were teachers of all the different disciplines (mathematics, informatics, physics, chemistry, and various engineering branches) and nationalities (French, Greek, Italian, and Spanish) and balanced by gender.

After the high-level introduction to symmetric and asymmetric cryptography, the didactical situation began. All three groups were given the same encrypted (with the PDS cryptosystem) message (Fig. 4) to decrypt in one hour, but starting with different information. The groups were also asked to pay attention to their solving strategies and keep track of difficulties and results so that they could later present their group's work to everyone (10 minutes of presentation and 5 of Q&As). English was used both for the interaction between participants and for presenting their groups' work.

One researcher was associated with each group to observe mainly the development of decryption strategies but also the use of the different disciplinary languages (from mathematics and informatics but also from the other disciplines of the group members) and the communication dynamics of the groups. The purpose of this observation was to verify whether the implementation of the situation had provided developments consistent with the a priori analysis to capture the students' interactions with different disciplinary and interdisciplinary concepts, objects, and methods between different semiotic

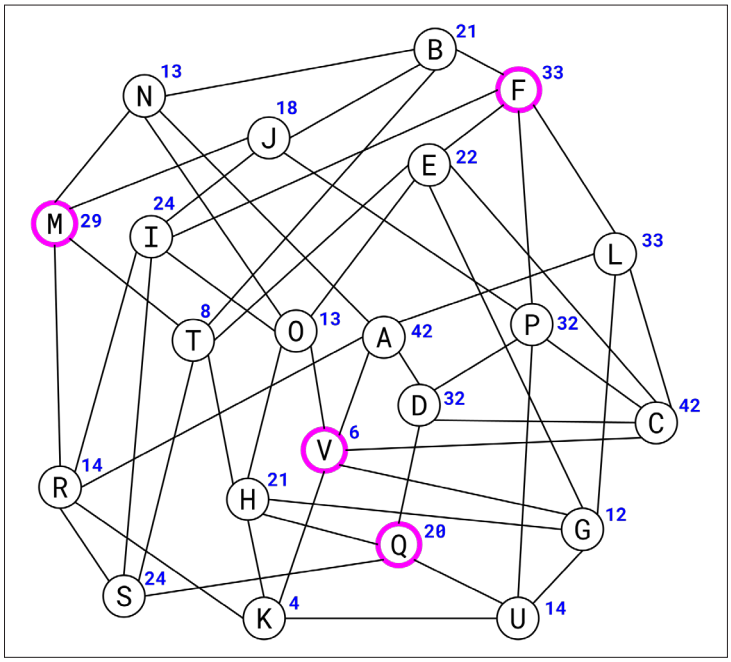

Fig. 4. The graph with the public values in blue (i.e., the encrypted message) chosen for our experimentation. In bold and magenta, the PDS of the graph (i.e., the private key), which is the set $\{F, M, Q, V\}$.

representations. To support the observation, the researchers relied on a grid, a product of the a priori analysis phase (also informed by the 2021 preliminary online experimentation). Such a grid helps the researchers observe for each group three main dimensions: group work and communication dynamics, strategies for solving the decryption problem, and linguistic and epistemological interdisciplinary elements. The grid is provided in Appendix A. In addition to the observations collected by the researchers, all the sessions were video-recorded.

An informed consent, explaining the objectives of the research, the data collection tools, and the commitment to process data in a pseudo-anonymous manner was provided and signed by all participants.

8. A Posteriori Analysis

The a posteriori analysis showed that the participants tried almost all of the problem-solving strategies we have described in the a priori analysis (Section 6). The strategies were not formulated by the student teachers exactly as they were presented in the a priori analysis. Nevertheless, the data collected showed that all groups tried (parts of) all the expected strategies. The participants' presentations and related discussions demonstrate their intuitive understanding of the strategies and why they are correct.

More specifically, group C was given the encrypted message and no other information (see Subsection 6.1 for the information given to each group). The participants solved the problem by formulating a system of linear equations (22 equations and 22 variables) and solving it with an online automatic linear solver[10]. The solver was suggested by the observing researcher only after the group had formulated the system of equations and decided that they would try to solve it. We hypothesise that the opportunity to use the software tool influenced their strategy to solve the problem. The size of this linear system makes it difficult to solve it by hand. With an automatic solver, the students could form and solve the system. Without an automatic solver available, they would probably have abandoned this strategy and tried to find a different one; for example, finding appropriate linear combinations of equations using the graph.

Group A was given the definition of *perfect dominating set* and the actual PDS on the graph, and they had to figure out how to use this information to decrypt the secret message. The researcher observing did not explain the definition of PDS; the participants had to figure out by themselves how to use the PDS to decrypt the secret message. This implies writing down the linear system by interpreting the encryption algorithm and trying to make the connection between the system and the PDS definition. We observed that understanding the PDS definition presents several difficulties. To begin with, the PDS definition is complex and structured in three steps. First, there is the notion of *domination*; second, the notion of *dominating set*; third, the notion of *perfect dominating set*. *Domination* is a symmetrical property (i.e., if node a dominates node b, then b also dominates a), while in natural language, domination is usually non-symmetric. Comprehending and using this definition was not easy for the students dealing with this property for the first time. In addition, domination was defined by the following: 'A vertex v of a graph G dominates vertex u if either $v = u$ or there is an edge from v to u.' We observed that 'if $v = u$' was not interpreted as 'vertex v dominates itself', as was intended. Instead, the participants thought this was related to the public values written on the nodes: their interpretation was that a vertex dominates the vertices with the same public value. This may be related to mathematics' different uses of the equality symbol. In mathematics, it is common to refer to length, width, and measure of quantities in phrases such as *the length of side v is 3 cm, and we write* $v = 3$ where the equality symbol is used to give the value 3 to the variable v, i.e., with a classical assignment meaning from an informatics perspective. We also use it to express the equality of values, so we can write $u = v$ if they have the same value. Moreover, the equality symbol in algebra is often used to express that two different letters (like u and v) refer to the same object, as we did in defining a PDS. Furthermore, although we thought that group A had the 'easiest' task since they had the private key at their disposal, it was revealed that their task was not trivial. Finding out how to use the PDS in order to decrypt requires three main steps of the PDS nodes:

1. Understanding the definition of PDS.
2. Formulating the linear system based on the encrypted message.

[10]Built by us (`https://lodi.ml/solver`) in Python with the SymPy library (`www.sympy.org`).

3. Translating the PDS properties (on the graph) into properties related to the equations.

The last step required a change of semiotic registers and was particularly tricky for the participants. We observed that they spent much time understanding the definition of the PDS and also formulating and reducing the linear system but had difficulty making the connection between the two.

Group B was given (i) the PDS definition and (ii) the decryption algorithm (given the PDS). As was expected, the participants started looking for the PDS in the graph. They partially tried all three algorithms presented in the a priori analysis. Note that all three algorithms are not polynomial: they are a 'structured' exhaustive search of the PDS in the graph. Therefore, the objective of the situation is not to solve the problem but to translate the PDS properties into algorithmic steps on the three different registers used (the graph, the lists, and the adjacency matrix). In that sense, the students succeeded at the task. More precisely, group B started with the list algorithm: they formed the list of neighbours for every node, they interpreted the PDS properties with the task of finding a number of lists with empty intersections whose union covers the graph, and they started comparing lists. In order to make a more efficient comparison, they decided to take into consideration the lists' size, too: given that the graph has 22 nodes, if they had a union of lists with size X, they only considered lists with a cardinal of size $\leqslant 22 - X$ for the following step. As a next step, instead of ordering the lists to facilitate the comparison, they decided to use a matrix, which led to forming the adjacency matrix of the graph. As one of the students stated, '[with a matrix] it is easier to see the connections [between nodes]'. Algorithm 1 came up in the last part of the situation in a different form. Instead of choosing a starting node (assumed in the PDS) and erasing its neighbours and its neighbours' neighbours, the students decided to note all paths of length 2 starting from a node (assumed in the PDS) and then all the paths of length 3, which would give them the possible candidates for the second node in the PDS. This idea was not followed because of limited time.

We observed that almost all participants took part in the conversation and in the final discussion about interdisciplinarity. All the groups developed the solving techniques and strategies that we had anticipated (not necessarily all of them). On the basis of our a priori analysis, we interpret this as a sign of appropriation of the problem and (at least partial) understanding of the main ideas behind the strategies. Although there were some language misunderstandings during the problem-solving process, we observed that during the discussion, the students were able to make proper use of the terminology used for graphs, cryptography, and linear system solution.

To conclude, the groups' work during the implementation supports our a priori analysis of the didactical situation. The participants were strongly involved in problem solving, and our observations (regarding the retroactions and the strategies used) indicate that our organisation of the *milieu* and our choices of didactical variables proved effective. Indeed, researchers needed very few limited interventions to support the groups' autonomous work. The work done in the different groups was generally consistent with the expectations of the a priori analysis.

9. Discussion

We presented the study of a didactical situation on cryptography between informatics and mathematics, designed, implemented, and analysed using the Theory of Didactical Situations within the Didactical Engineering methodology.

The implementation of the Didactical Situation showed a learning potential for fundamental concepts, methods, and ideas not only of cryptography but also of mathematics and informatics. Given the nature of the problem-solving activity, the participants need to find and elaborate strategies in order to solve the problem (RP1). To elaborate these strategies, they need to explore and understand (at least intuitively) several concepts and methods from mathematics and informatics (RP2), such as backtracking, LIFO stacks, the adjacency matrix of a graph, matrices for modelling a linear system of equations, and also move between semiotic registers by interpreting the properties of the PDS definition written in set theory language, graphically, or by using lists. The choices of the values for the didactical variables are essential in this sense: for example, the graph (its size, the vertex degrees and its graphical representation) does not allow the participants to find the PDS by trial and error, and reveals the hardness of the problem while still allowing them to form a linear system by hand; also, the number of PDSs in the graph is closely related to the existence of a (unique) solution of the linear system. The participants are restricted in their retroactions by those elements and, therefore, need to explore the strategies in order to solve the problem.

Moreover, we conjecture that our didactical situation has the learning potential to introduce topics like the complexity and correctness of algorithms, as well as to work on graphs, dominating sets, linear systems, and matrices and their representations in mathematics and informatics. To illustrate that, we paraphrase some questions the participants discussed while trying to solve the problem: Is there always a perfect dominating set in a graph? And a dominating set? How complex is it to solve a linear system? Are the graph algorithm and the list algorithm more efficient than the brute force solution? What is the relationship between the linear system and the PDS? Why is decoding with a PDS correct?

9.1. *Future Work*

The research work of the IDENTITIES project of which the current work is part is still ongoing. In this paper, we have analysed our observations of the implementation of the didactical situation on cryptography. We conjecture that students were able to grasp the challenges of public-key cryptography and develop a better understanding of the interdisciplinary objects involved. Nevertheless, we still have to refine this analysis rigorously: transcribe the audio and process the videos in order to identify and analyse all the steps of the problem-solving procedure in detail. Next, we have to identify all interdisciplinary boundary objects that come into play and analyse them from an interdisciplinary point of view, for example, by using the Akkerman and Bakker (2011) framework on interdisciplinarity.

A future direction for this work would be to implement our didactical situation with different values for the didactical variables and also to adapt and test the activity with high school students.

9.2. *Conclusions*

To conclude, we successfully used the Didactical Engineering research methodology to design a teaching activity that enables students to explore the idea and the complexity of public-key cryptosystems while interacting with the informatics and mathematics interdisciplinary objects involved in that activity and the related disciplinary concepts. Therefore, to overcome the obstacles students encounter in this didactical situation, they must mobilise concepts, methods, and practices of mathematics and informatics, moving between semiotic representations of interdisciplinary objects.

The specific analysis of the interdisciplinary interactions between pre-service STEM teachers and the model of interdisciplinarity that can emerge from this kind of activity is part of the larger project and will be analysed in future works.

Acknowledgments

We deeply thank all the participants of the two IDENTITIES summer schools.

Funding

This work was supported by:

- IDENTITIES Project, co-funded by the Erasmus+ Programme of the European Union under Grant Agreement n° 2019-1-IT02-KA203-063184 (all authors);
- Spoke 1 “FutureHPC&BigData” of the Italian Research Center on High-Performance Computing, Big Data and Quantum Computing (ICSC) funded by MUR Missione 4 Componente 2 Investimento 1.4: Potenziamento strutture di ricerca e creazione di “campioni nazionali di R&S (M4C2-19)” – Next Generation EU (NGEU) (Lodi);
- SERICS (PE00000014) under the Italian MUR National Recovery and Resilience Plan funded by the European Union – NextGenerationEU (Martini);
- INdAM GNSAGA (Martini).

References

Akkerman, S.F., Bakker, A. (2011). Boundary Crossing and Boundary Objects. *Reviewof EducationalResearch*, 81(2), 132–169. https://doi.org/10.3102/0034654311404435

Anane, R., Alshammari, M.T. (2020). A Dynamic Visualisation of the DES Algorithm and a Multi-Faceted Eval-

uation of Its Educational Value. In: *Proceedings of the 25th ACM Conference on Innovation & Technology in Computer Science Education*. ITiCSE '20. ACM, New York, NY, USA, pp. 370–376. 9781450368742. https://doi.org/10.1145/3341525.3387386

Artigue, M. (1994). Didactical engineering as a framework for the conception of teaching products. *Didactics of mathematics as a scientific discipline*, 13, 27–39.

Artigue, M. (2020). Didactic Engineering in Mathematics Education. In: Lerman, S. (Ed.), *Encyclopedia of Mathematics Education*. Springer International Publishing, Cham, pp. 202–206. 9783030157883 9783030157890. https://doi.org/10.1007/978-3-030-15789-0_44

Barelli, E., Barquero, B., Romero, O., Aguada, M.R., Giménez, J., Pipitone, C., Sala-Sebastià, G., Nipyrakis, A., Kokolaki, A., Metaxas, I., Michailidi, E., Stavrou, D., Bartzia, I., Lodi, M., Sbaraglia, M., Modeste, S., Martini, S., Durand-Guerrier, V., Bagaglini, V., Satanassi, S., Fantini, P., Kapon, S., Branchetti, L., Levrini, O. (2022). Disciplinary identities in interdisciplinary topics: challenges and opportunities for teacher education. In: Carvalho, G.S., Afonso, A.S., Anastácio, .Z. (Eds.), *Proceedings of ESERA 2021*. European Science Education Research Association 2021 Biannual Conference*: Vol.* 13, pp. 934–943. https://esera2021.org/en/content/eproceedings/conference-proceedings/conf-proceedings.html

Bell, T., Witten, I., Fellows, M. (2015). Public Key Encryption. CS Unplugged. https://classic.csunplugged.org/activities/public-key-encryption/

Bell, T., Thimbleby, H., Fellows, M., Witten, I., Koblitz, N., Powell, M. (2003). Explaining cryptographic systems. *Computers & Education*, 40(3), 199–215. https://doi.org/10.1016/S0360-1315(02)00102-1

Brousseau, G., Warfield, V. (2020). Didactic Situations in Mathematics Education. In: Lerman, S. (Ed.), *Encyclopedia of Mathematics Education*. Springer International Publishing, Cham, pp. 206–213. 9783030157883 9783030157890. https://doi.org/10.1007/978-3-030-15789-0_47

Brousseau, G., Sarrazy, B., Novotná, J. (2020). Didactic Contract in Mathematics Education. In: Lerman,

S. (Ed.), *Encyclopedia of Mathematics Education*. Springer International Publishing, Cham, pp. 197–202. 9783030157883 9783030157890. https://doi.org/10.1007/978-3-030-15789-0_46

Brown, C., Crabbe, F., Doerr, R., Greenlaw, R., Hoffmeister, C., Monroe, J., Needham, D., Phillips, A., Pollman, A., Schall, S., Schultz, J., Simon, S., Stahl, D., Standard, S. (2012). Anatomy, Dissection, and Mechanics of an Introductory Cyber-Security Course's Curriculum at the United States Naval Academy. In: *Proceedings of the 17th ACM Conference on Innovation & Technology in Computer Science Education*. ITiCSE '12. ACM, New York, NY, USA, pp. 303–308. 9781450312462. https://doi.org/10.1145/2325296.2325367

CECE (The Committee on European Computing Education) (2017). Informatics Education in Europe: Are We All In The Same Boat? Technical report, Informatics Europe & ACM Europe. 9781450353618. https://doi.org/10.1145/3106077

Chevallard, Y., Bosch, M. (2020). Anthropological Theory of the Didactic (ATD). In: Lerman, S. (Ed.), *Encyclopedia of Mathematics Education*. Springer International Publishing, Cham, pp. 53–61. 9783030157883 9783030157890. https://doi.org/10.1007/978-3-030-15789-0_100034

Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C. (2022). *Introduction to algorithms* (4th ed.). The MIT Press, Cambridge, Massachusetts. 9780262046305.

CSTA (2017). CSTA K-12 Computer Science Standards, rev. 2017. Technical report, Computer Science Teachers Association. http://www.csteachers.org/standards

Deshpande, P., Lee, C.B., Ahmed, I. (2019). Evaluation of Peer Instruction for Cybersecurity Education. In: *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. SIGCSE '19. ACM, New York, NY, USA, pp. 720–725. 9781450358903. https://doi.org/10.1145/3287324.3287403

Duval, R. (1995). *Sémiosis et pensée humaine: registres sémiotiques et apprentissages intellectuels*. Peter Lang, Berne.

Duval, R. (2017). *Understanding the Mathematical Way of Thinking – The Registers of Semiotic Representations*. Springer International Publishing, Cham. 9783319569093 9783319569109. https://doi.org/10.1007/978-3-319-56910-9

Fees, R.E., da Rosa, J.A., Durkin, S.S., Murray, M.M., Moran, A.L. (2018). Unplugged cybersecurity: An approach for bringing computer science into the classroom. *International Journal of Computer Science Education in Schools*, 2(1), 3–13. https://doi.org/10.21585/ijcses.v2i1.21

Fellows, M.R., Hoover, M.N. (1991). Perfect domination. *Australas. J Comb.*, 3, 141–150.

Fellows, M.R., Koblitz, N. (1994). Combinatorially based cryptography for children (and adults). *Congressus Numerantium*, 99, 9–41.

Haynes, T.W., Hedetniemi, S., Slater, P. (2013). *Fundamentals of domination in graphs*. CRC press, Boca Raton.

Joint Task Force on Cybersecurity Education (2018). *Cybersecurity Curricula 2017: Curriculum Guidelines for Post-Secondary Degree Programs in Cybersecurity*. ACM, New York, NY, USA. 9781450389198. https://dl.acm.org/doi/book/10.1145/3184594

Katz, J., Lindell, Y. (2007). *Introduction toModernCryptography*. Chapman and Hall/CRC, New York. https://doi.org/10.1201/9781420010756

Klostermeyer, W.F. (2015). A Taxonomy of Perfect Domination. *Journal of Discrete Mathematical Sciences and Cryptography*, 18(1–2), 105–116. https://doi.org/10.1080/09720529.2014.914288

Konak, A. (2014). A cyber security discovery program: Hands-on cryptography. In: *2014 IEEE Integrated STEM Education Conference*. IEEE, New York, NY, USA. https://doi.org/10.1109/ISECon.2014. 6891029

Konak, A. (2018). Experiential Learning Builds Cybersecurity Self-Efficacy in K-12 Students. *Journal of Cyber-security Education, Research and Practice*, *2018*(1). https://digitalcommons.kennesaw.edu/jcerp/vol2018/iss1/6

Lindmeier, A., Mühling, A. (2020). Keeping Secrets: K-12 Students' Understanding of Cryptography. In: *Proceedings of the 15th Workshop on Primary and Secondary Computing Education*. WiPSCE '20. ACM, New York, NY, USA. 9781450387590. https://doi.org/10.1145/3421590.3421630

Ma, J., Tao, J., Mayo, J., Shene, C.-K., Keranen, M., Wang, C. (2016). AESvisual: A Visualization Tool for the AES Cipher. In: *Proceedings of the 21st ACM Conference on Innovation & Technology in Computer Science Education*. ITiCSE '16. ACM, New York, NY, USA, pp. 230–235. 9781450342315. https://doi.org/10.1145/2899415.2899425

Miani, L. (2021). *Highlighting interdisciplinarity between physics and mathematics in historical papers on special relativity: design of blended activities for pre-service teacher education*. Master thesis in Physics, University of Bologna. http://amslaurea.unibo.it/23544/

Millar, V. (2020). Trends, Issues and Possibilities for an Interdisciplinary STEM Curriculum. *Science & Education*, 29(4), 929–948. https://doi.org/10.1007/s11191-020-00144-4

Modeste, S. (2016). Impact of Informatics on Mathematics and Its Teaching. In: Gadducci, F., Tavosanis, M. (Eds.), *History and Philosophy of Computing*. Springer International Publishing, Cham, pp. 243–255. 978-3-319-47286-7.

Pino-Fan, L., Guzmán, I., Duval, R., Font, V. (2015). The theory of registers of semiotic representation and the onto-semiotic approach to mathematical cognition and instruction: linking looks for the study of mathematical understanding. In: *Proceedings of the 39th Conference of the International Group for the Psychology of Mathematics Education* (Vol. 4). PME, Hobart, Australia, pp. 33–40.

Rosamond, F. (2018). Computational Thinking Enrichment: Public-Key Cryptography. *Informatics in Education*, 17(1), 93–103. https://doi.org/10.15388/infedu.2018.06

Schweitzer, D., Brown, W. (2009). Using Visualization to Teach Security. *Journal of Computing Sciences in Colleges*, 24(5), 143–150.

Simms, X., Chi, H. (2011). Enhancing Cryptography Education via Visualization Tools. In: *Proceedings of the 49th Annual Southeast Regional Conference*. ACM-SE '11. ACM, New York, NY, USA, pp. 344–345. 9781450306867. https://doi.org/10.1145/2016039.2016139

Skiena, S.S. (2020). *Combinatorial Search*. Springer International Publishing, Cham, pp. 281–306. 978-3-03054256-6. https://doi.org/10.1007/978-3-030-54256-6_9

Sommers, J. (2010). Educating the next Generation of Spammers. In: *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*. SIGCSE '10. ACM, New York, NY, USA, pp. 117–121. 9781450300063. https://doi.org/10.1145/1734263.1734302

STEM Task Force (2014). Innovate: a blueprint for science, technology, engineering, and mathematics in California public education. Technical report, Californians Dedicated to Education Foundation, Dublin, CA, USA. https://www.cde.ca.gov/pd/ca/sc/documents/innovate.pdf

Turner, C.F., Taylor, B., Kaza, S. (2011). Security in Computer Literacy: A Model for Design, Dissemination, and Assessment. In: *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*. SIGCSE'11. ACM, New York, NY, USA, pp. 15–20. 9781450305006. https://doi.org/10.1145/ 1953163.1953174

Švábenský, V., Vykopal, J., Čeleda, P. (2020). What Are Cybersecurity Education Papers About? A Systematic Literature Review of SIGCSE and ITiCSE Conferences. In: *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. SIGCSE '20. ACM, New York, NY, USA, pp. 2–8. 9781450367936. https://doi.org/10.1145/3328778.3366816

Wilson, C., Sudol, L.A., Stephenson, C., Stehlik, M. (2010). *Running on Empty: The Failure to Teach K–12 Computer Science in the Digital Age*. Association for Computing Machinery, New York, NY, USA. 9781450388672.

E.-I. Bartzia (BS and MS in Mathematics, Ph.D. in CS). After a PhD in the formalisation of cryptographic algorithms at INRIA, she taught mathematics at the University of Paris 8. Since 2019 she has been working as a post-doctorate researcher on the didactics of mathematics and interactions with computer science. Her research interests focus on the teaching of proof and proving, the use of proof assistants for teaching and learning mathematics, problem solving, cryptography and algorithmic thinking.

M. Lodi (BS, MS, Ph.D. in CS) is a Junior Assistant Professor at the Department of Computer Science and Engineering, University of Bologna, Italy. His research focuses on Informatics Education: he studies historical, epistemological, didactical, and motivational aspects of introducing Informatics (and programming in particular) in K-12 education, especially with constructivist approaches. Recently, he is also interested in the didactics of other CS topics like cryptography, high-performance computing, and quantum computing.

M. Sbaraglia (BS, MS, Ph.D. in Informatics Engineering). After researching biochemically inspired self-organizing systems, he worked for a few years as a developer. He has been teaching Informatics in high school since 2013 and has been holding a chair since 2016. He defended his thesis on Informatics Education in 2023 at University of Bologna. He researches introductory programming, CS1 online learning, big ideas of Cryptography and interdisciplinarity between Informatics and Mathematics.

S. Modeste (Ph.D. in didactics of mathematics, Grenoble, 2012) is an assistant professor at the University of Montpellier, in didactics of mathematics and computer science. His Ph.D. was on the topic of the teaching and learning of algorithmics in mathematics. Since then, most of his research deals with mathematics and computer science interactions in a didactical perspective, teaching and learning discrete mathematics, algorithmics and programming in mathematics education.

V. Durand-Guerrier (PhD in didactics of Mathematics, Lyon, 1996) is emeritus professor at the University of Montpellier (France) in Didactics and Epistemology of Mathematics. After a position as assistant professor at the Institute for teacher training in Lyon, she moved in 2009 as full professor to the University of Montpellier; Her main research interest is in the relationships between logic, language, argumentation, proof in the teaching and learning of mathematics from primary school to university.

S. Martini (Ph.D. in Computer Science, Pisa, 1987) is professor of Computer Science at University of Bologna. Before joining University of Bologna in 2002, he taught at the universities of Pisa and Udine. He has been a visiting scientist at the former Systems Research Center of Digital Equipment Corporation, Palo Alto; at Stanford University; at École normale supérieure, Paris; at Université Paris 13; at University of California at Santa Cruz; and the Collegium – Lyon Institute for Advanced Studies. His research interests are in the logical foundations of programming languages, in history and philosophy of computer science, and in computer science education.

Appendix A. Grid

Observer <*Name*> **Start time** <*hh:mm*>
Group <*ID*> (<*milieu short description*>) **End time** <*hh:mm*>

Participants (prospective teachers)

- **P1** <*Name Surname*> – <*discipline*>
- **P2** <*Name Surname*> – <*discipline*>
- **P3** <*Name Surname*> – <*discipline*>
- **P4** <*Name Surname*> – <*discipline*>
- **P5** <*Name Surname*> – <*discipline*>

Group work						
	P1	**P2**	**P3**	**P3**	**P5**	*Notes*
Works mainly together with the group						
Works in subgroups *specify subgroups & different approaches*						
Works alone *specify approach(es)*						
Constantly communicates with others						
Does not (try to) communicate						
Not understood by other students						
Other / notes:						
Problem-solving strategies						
	P1	**P2**	**P3**	**P3**	**P5**	*Notes*
Heuristic algorithm (partitions of the graph)						
proposed						
followed						
abandoned						
Algorithm using lists (each list is a neighborhood of a PDS node)	*(alt. desc.: find a subset of lists of neighbors that include each node exactly once)*					
proposed						
followed						
abandoned						

	P1	**P2**	**P3**	**P3**	**P5**	*Notes*
Algorithm 1 using the adjacency matrix: find a linear combination that has 1 everywhere	*(alt. desc.: use the adjacency matrix to represent the graph and then to find a subset of rows whose sum is exactly a list of 1s)*					
proposed						
followed						
abandoned						
Algorithm 2 using the adjacency matrix: create a linear system of equations (to be solved by a linear solver)						
proposed						
followed						
abandoned						
Other strategy (specify):						
proposed						
followed						
abandoned						
Other / notes:						

Boundary objects						
	P1	**P2**	**P3**	**P3**	**P5**	*Notes*
Talks about the adjacency matrix as a way to represent and solve a system of equations						
Talks about the adjacency matrix as a way to represent the graph (1 for neighbor)						
Other / notes:						

Linguistic aspects						
	P1	**P2**	**P3**	**P3**	**P5**	*Notes*
Talks about the problem using a cryptography language, e.g., PDS as the (private) key						
Talks about the problem using own discipline language						
Talks about the problem using OTHER discipline languages						
Other / notes:						
Interdisciplinary aspects						
	P1	**P2**	**P3**	**P3**	**P5**	*Notes*
Effort in being understood by students of other disciplines						
Remain in / go back to their “disciplinary confort zone”						
Other / notes:						
Disciplinary thinking / acting						
	P1	**P2**	**P3**	**P3**	**P5**	*Notes*
"First make it work, then make it nice" (CS) approach						
Comfort in doing a lot of computations (CS) (e.g., a lot of equations without simplifying, brute force for sublists)						
Search for an elegant representation/strategy before trying to solve (Math/Phys?)						
Discomfort in doing a lot of computations (search for simplification/abstraction) (Math/Phys?)						
Other / notes:						