Cryptography as a field to foster interactions between mathematics and informatics, and algorithms. Analysis of a didactical situation

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

Cryptography as a field to foster interactions between mathematics and informatics, and algorithms. Analysis of a didactical situation

Simon Modeste[1], Evmorfia-Iro Bartzia[1], Michael Lodi[2], Marco Sbaraglia[2], Viviane Durand-Guerrier[1] and Simone Martini[2]

[1]IMAG, University of Montpellier, CNRS, Montpellier, France; simon.modeste@umontpellier.fr

[2]Alma Mater Studiorum Università di Bologna, Italy, & Laboratorio Nazionale CINI "Informatica e Scuola", Rome, Italy

We present the conception and analysis of a situation dealing with the principles of public-key cryptography and aiming at exploring informatics and mathematical concepts and methods. We rely on the Theory of Didactical Situations to design a situation (based on an unplugged activity) about public-key cryptography using graphs. After the preliminary analysis of the content, we conceived a didactical situation and developed its a priori analysis. The description of the associated solving strategies illustrates the interplay between mathematics and informatics, and the role of algorithms and algorithmic thinking.

Keywords: Public-key cryptography, computer science unplugged, mathematics and informatics, didactical engineering, theory of didactical situations.

Introduction

Mathematics and Informatics have "strong links and a common history", they have common backgrounds (e.g. logic), concepts (e.g. graphs, functions), and "fields developing at their interface", and "a very similar relation to other sciences through modelling and simulation" (Modeste, 2016, pp. 243-244). They are two autonomous and distinct disciplines, but their frontier is blurry, with mutual contributions. From an educational perspective, we consider it important to foster interdisciplinary teaching to make students aware of threads that both disciplines share.

In the context of the IDENTITIES Project (https://identitiesproject.eu/), about interdisciplinarity in STEM education and pre-service teacher training, we have chosen to explore possibilities offered by cryptography. Our choice was motivated by the epistemological reason that cryptography is a deeply interconnected field between informatics and mathematics, involving many concepts from the two disciplines, but also "boundary objects" (Akkerman and Bakker, 2011) belonging to both.

We hypothesize that the research methodology offered by the Theory of Didactical Situations (Brousseau & Warfield, 2020) is relevant to design didactical situations at the interface of mathematics and informatics.

Our main research question is "How cryptography can foster mathematics-informatics interactions and algorithms, and what kind of learning activity can it generates?".

For exploring this question, we developed a didactical situation based on an activity on public-key cryptography as a central part of a module for pre-service teacher training. We present here the didactical situation that, in our view, can be adapted for various audiences: in-service teachers, high school students, undergraduates, PhD students, both from informatics or mathematics background.

After introducing the Theory of Didactical Situations, we will present the didactical situation, and its *a priori* analysis, which shows how the situation and its organization foster interactions between mathematics and informatics, and the role of algorithms and algorithmic thinking.

Theory of Didactical Situations

The Theory of Didactical Situations (TDS) offers conceptual tools for the design of teaching and learning Situations. Students' learning is seen as the result of interactions in a system of relationships between students, a teacher and a milieu. The TDS core-conceptual tools that we will use in this paper are the following: the concepts of *milieu* and of *adidacticity,* and the notion of *didactic variables (Brousseau & Warfield, 2020).* The *milieu* of a situation is composed of the set of material objects, the available knowledge, and interactions with teacher and students. In a didactical situation, students rely on their available knowledge to engage in actions to solve the problem at stake. During this activity, the *milieu* provides *retroactions*. *Adidacticity* characterizes the phases in which students are able to interpret autonomously the retroactions of the milieu. A *didactical variable* is a variable of a situation for which changing the value may impede the solving strategies (validity, complexity) and their hierarchy, and on which the teacher can act according to his objectives. Identifying the *didactical variables* and their effects is the core of the *a priori* analysis.

Presentation of the didactical situation

Our didactical situation aims both at teaching the "core idea" (Lodi et al., 2022) of public-key cryptography and making participants interact with mathematical and informatics objects. We based on a public-key cryptography problem (Fellows, 1994) which uses a cryptosystem leveraging on the computationally hard problem of finding a *perfect dominating set* on a graph.

Elements of preliminary analysis

While the importance of teaching cryptography has been recognized in both graduate and K-12 curricula (Joint Task Force on Cybersecurity Education, 2018; CSTA, 2017), often, it is treated as just one of the many topics of cybersecurity or with a too technical and instrumental focus rather than on its core ideas (for recent reviews, see Švábenský et al. (2020) and Lodi et al. (2022)). Research shows, however, that hands-on, cooperative, and inquiry-based activities can improve students' self-efficacy and problem-solving skills (Konak, 2018). Some authors proposed unplugged activities for students to experience cryptographic algorithms, protocols and attacks (e.g., Bell et al., 2003; Konak, 2014). Moreover, communicating in secret and trying to decrypt messages without knowing the key is not only engaging and motivating for students (Lindmeier and Mühling, 2020), it has also, from a didactical perspective, a strong potential for *adidacticity*, allowing self-directed learning through autonomous engagement with the task (e.g., when you can easily verify if you have well decrypted a message or if your (de)crypting programs run correctly).

A public-key cryptosystem using perfect dominating sets on graphs

An *encryption scheme* allows for confidential communication between two parties over a public channel. A *plaintext message* is transformed into a *ciphertext* (i.e., an encrypted message) using an *encryption algorithm*, the security of which is dependent on one or more *keys*. There are two types of cryptosystems: *symmetric* (or *secret-key*) and *asymmetric* (or *public-key*). In a symmetric

cryptosystem, the same key is used for both encryption and decryption. In an asymmetric cryptosystem, a different key is used for encryption (public key) and decryption (private key).

In our case, we considered a public-key cryptosystem. The important components of a public-key encryption scheme include: a key generation algorithm (Gen) that creates a pair of keys (pk, sk) (public key and private key) for each user, an encryption algorithm (Enc) that converts a plaintext message m into a ciphertext $c = Enc_{pk}(m)$ using the recipient's public key, and a decryption algorithm (Dec) that converts the ciphertext back into the original plaintext using the recipient's private key ($m = Dec_{sk}(c)$). Enc and Dec algorithms must be computationally efficient when the corresponding keys are known, and the scheme's security is based on the difficulty of computing the Dec function without having access to sk.

An asymmetric cryptosystem that leverages on the computationally difficult problem called Perfect Dominating Set (PDS in the following) problem has been proposed (Fellows and Koblitz, 1994). Given a graph $G = (V, E)$ with a set of vertices V and a set of edges E, a *(closed) neighbourhood* of a vertex $u \in V$ is the set $N[u] = \{v \in V / uv \in E\} \cup \{u\}$ (that is, the vertices in V that are adjacent to u, and u itself). 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 in S. If S is a dominating set of G, then every vertex in V is a neighbour to at least one vertex in 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 dominating set* (often referred to also as *perfect code*). Figure 1 (left) gives an example of a graph with a PDS. Thus, the PDS problem is the following (Fellows and Hoover, 1991): given a graph $G = (V, E)$, output a PDS of G, if one exists.

Deciding whether a graph has a PDS is, in general, NP-complete (Klostermeyer, 2015, p. 107). Therefore, our PDS Problem is NP-hard, which means we only have algorithms that take exponential time in the number of nodes, and we may never be able to improve that. This feature can be used to design a cryptosystem, as we will see.

The PDS problem can be used to develop an asymmetric cryptographic system because (1) starting from a set of vertices S, it is possible to construct a graph that has S as a PDS; but (2) given a graph that has a PDS, it is hard to find it by only knowing the graph.

The PDS cryptosystem works as follows. Bob wants to confidentially communicate a secret message (an integer) m to Alice. They can use the following encryption protocol.

- Alice creates a graph $G = (V, E)$ (where $V = \{v_1, v_2, \ldots v_k\}$) that has S as PDS. G is Alice's public key, and the PDS S is Alice's private key.
- Bob writes m as the sum of random integers $m_1, m_2, \ldots, m_k$ ($m_1 + m_2 + \ldots + m_k = m$)
- Bob assigns an m_i to each vertex v_i of V. m_i is now called the secret value v_i.
- Bob calculates a public value p_i of each vertex v_i by summing the secret values of v_i's closed neighbourhood (i.e. including v_i own secret value).
- Bob creates the encrypted message by writing on each vertex v_i of the graph its public value p_i (and removing its secret value m_i).

Figure 1 (right) shows the previous graph with public (grey) and secret values (black underlined).

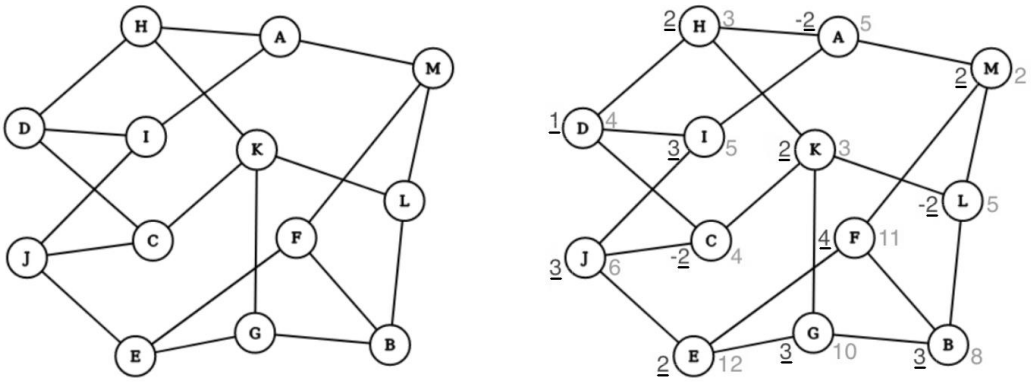

Fig. 1 (left): The subset {I, K, F} is a PDS for the graph ; (right): The secret message $m = 19$ encrypted using the same graph. Secret values in black underlined, and public values in grey.

To decrypt the encrypted message (the graph G with public values) that Bob sent her, Alice calculates the sum of the public values of the nodes in the PDS (the PDS is her private key).

In principle, the graph G (public key) and the encrypted message can freely circulate: a priori, the system is secure because finding the PDS from the graph is computationally challenging (NP-hard). However, algebraic attacks can be performed (Fellows & Koblitz, 1994), which makes it not a good cryptosystem in real life, but very interesting for a didactical situation.

While the cryptosystem is known (Fellows and Koblitz, 1994) and has been used in teaching activities (e.g., Bell et al., 2003), our main contribution is the design of an original didactical situation around it, with an organization of its milieu, an identification of the didactical variables, and an detailed a priori analysis, that foster its learning potential in mathematics and informatics.

In our didactical situation, participants are given a complex task (decrypting a secret message encrypted with the PDS cryptosystem). Due to the careful choice of didactical variables, participants have to devise solving strategies that involve understanding and applying concepts and methods from mathematics and informatics and sometimes the change of semiotic registers.

The didactical situation and elements of the *a priori* analysis

The objectives of the didactical situation are **(1)** Introduce some general concepts and terminology from cryptography (e.g., plaintext and ciphertext, encryption and decryption algorithms, attack models, private and public keys, difficult-to-reverse problem, one-way function) and make students understand and explore the ideas and challenges of public-key cryptography; **(2)** Make students explore and interact with mathematical and informatics concepts and interdisciplinary objects (e.g., graphs, algorithms, matrices).

The didactical situation is organised as follows:

Phase 1: *Encryption*. Participants are shown the encryption algorithm using the graph *G* (the public key). They are neither taught what a PDS is (not needed for encryption) nor that *G* has a PDS.

Phase 2: *Cryptanalysis*. Participants are divided into three groups. The same encrypted message (i.e., the graph *G* with public values) is given to everyone, and they are asked to decrypt it. Each group is given different information to solve the problem, as detailed in the following.

The aim of making three groups, sharing and debating their results after the research phase, is to make them grasp the issues of asymmetric cryptography, the role of the public and secret keys, and that attacks can be done with different levels of information about the protocols. For each group, we present the information given and the position in which students are put. Then we describe the main strategies, highlighting mathematical, informatics, and algorithmics contents and thinking involved.

Group A – *Available information:* the PDS definition and a PDS for the graph G. No information on using the PDS to decrypt is given: they should find by themselves the decryption algorithm using the PDS. They are put in the position of engineers trying to design a decryption protocol.

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 graph *G*. The neighbourhoods can be represented as lists of vertices or graphically as 'stars' on the graph. By the cryptosystem construction, 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. In order to elaborate this strategy, it is needed to interpret the definition of PDS (expressed using terminology from set theory) on the graphical representation of the graph, and make the connection with the encryption procedure. More precisely, it is needed 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, i.e., the decryption of an encrypted message returns the plaintext message: $Dec_{sk}\left(Enc_{pk}(m)\right) = m$. This can involve making the decryption algorithm explicit and proving the encryption's correctness.

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 it. This incites the group to try to find the private key (the PDS) using the encrypted message and the public key. Group B is thus confronted with an instance of the difficult problem of finding a PDS in a graph. They are put in the position of attackers who know completely the protocols but not the private key.

Strategies: we describe three possible strategies, using different semiotic registers (Duval, 2017): the graph representation, the lists of vertices, and the graph's adjacency matrix. These three strategies amount to a structured, exhaustive search of the subsets of vertices to find the PDS.

Strategy 1: Finding stars. Given a graph *G = (V, E),* we have to find a set *S* that is a PDS of *G*.

Let *v* be a vertex of *V*. By the PDS definition, 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: i) the neighbouring vertices of *u* do not belong to *S*, and ii) 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 in *S*, we can deduce that its neighbours and the neighbours of its neighbours are not in *S*.

We iteratively add vertices to a set S to find a PDS. If we do not succeed, we backtrack to the vertices' choices to continue exploring potential PDSs. In informatics, *backtracking* is a 'systematic way to

run through all the possible configurations of a search space', especially useful when 'we must generate each possible configuration exactly once' (Skiena, 2020, p. 281): 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 and then interpreting the definition on the graphical representation of the graph. Systematising the steps of the algorithm requires an understanding of both the properties of domination and perfect domination and a intuitive idea of backtracking.

Strategy 2: Lists. For each vertex of *G*, we write its neighbourhood as a list of neighbour vertices. We then study these lists in order to find a set of lists whose intersection is empty and whose union covers graph *G*. The basic idea of this strategy is that each vertex of graph *G* belongs to exactly one neighbourhood of a vertex of *S*. The idea is to incrementally build a collection *L* of lists, such that their intersection is empty, while their union contains all the vertices of *G*. More technically, *L* is a LIFO stack, a data structure that implements the *LIFO* (last-in, first-out) policy: the last list added on top of L is the first to be removed when it is not suitable for building the PDS *S*. Elaborating this strategy requires understanding the perfect domination properties, expressing these properties using lists, and an intuitive understanding of a LIFO stack (even if not recognised as such).

Strategy 3: Adjacency matrix of the graph. This strategy consists in writing the adjacency matrix of the graph *G*: for a vertex *i*, in the corresponding row $l_i = [a_{i1}, a_{i2} \ldots, a\ ]$ the coefficients $a_{ij} = 1$ if the vertices *j* and *i* are connected and 0 otherwise. Note that here $a_{ii} = 1$ for all vertex *i* (because, in the PDS definition, we are considering closed neighbourhoods). If we find a set of rows whose sum is [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). This idea is very close to Strategy 2: we go through the set of rows of the matrix, including or excluding rows, to find a subset of rows whose sum is [1, 1, ..., 1] - but the register of representation is different. Elaborating this strategy requires understanding the properties of the PDS and expressing these properties using the adjacency matrix.

Group C – *Available information:* no information other than the encrypted message. They only know the encryption algorithm. This is expecting them to try to break the system without searching for the key, but exploiting other vulnerabilities. Thus, they are put in the position of attackers who do not try to solve the PDS problem, but explore other approaches to decrypt the message.

Strategy: Starting from the encrypted message, a linear system can be constructed as follows: for each vertex *v*, of public value p_v and neighbourhood $N[v] = [v, v_1, \ldots v_k]$, write the equation $x_v + x_{v_1} + \ldots + x_{v_k} = p_v$ 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. The linear system of those equations will have as many equations and unknowns as vertices in *G*. The solution of the linear system is the tuple of all secret values $[x_1, x_2, \ldots, x_n]$, whose sum is the plaintext message *m*. Note that, in this case, 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 created. Unfolding this strategy requires interpreting the cryptosystem as a linear system and examining its resolution. Note that the solution to the problem

does not necessitate the resolution of the linear system but just finding the sum of all secret values; among others possibilities, this can be done by finding the rows corresponding to the PDS nodes.

Principal didactical variables

The principal didactical variables identified, and their values for our learning objectives, are:

- Access to information: it is our main variable since each group has different information.
- The type of graph: it should be hard to find the PDS. Certain types of graph, for which it is known that the PDS problem is not hard (e.g., trees (Klostermeyer, 2015, p. 107)), should be excluded. Moreover, while the PDS problem is hard for planar graphs, we observed that using non-planar graphs makes the problem visually more difficult for the participants.
- The size of the graph: it should be large enough so that an exhaustive search of the PDS is tedious, but small enough so that writing the linear system is still be feasible by hand.
- The graph's maximum degree and the degrees of the vertices: a too visible difference of degrees between the vertices could influence the starting point and Strategy 1 (starting with vertices with higher or lower degree). The graph should be "almost regular" (but not with all nodes of same degree *k*, as in this case you can get the plaintext message without the PDS).

Conclusion: learning potential and links with algorithms

We have presented elements of the *a priori* analysis of a didactical situation in cryptography, highlighting the potential of cryptography to deal with contents in mathematics, informatics, and their links. A first observation from our analysis, is that the frontier between informatics and mathematics is blurry, and many concepts and ways of reasoning are shared at their interface. Among these concepts, we see that many algorithms or drafts of algorithms are at stake: in the asymmetric cryptography principle, as it is based on the notion of algorithmic problem complexity, but also through the notions and procedures involved in the solving. We see also that the algorithms support most of the reasoning in the strategies. We consider this as *algorithmic thinking*, in the sense of reasoning with and about algorithms (designing effective procedures to solve problems, formalizing and proving algorithms, use and combine them as tools in exploring an solving problems). One key point is that these learning potentials, in this cryptography problem, can only be realized with a careful organization of the didactical situation, based on a detailed *a priori* analysis.

We have experimented our situation in various contexts (science teachers education, mathematics students, in-service teachers education...) and data collected is under analysis. We have already noticed a strong stability of strategies developed in these different contexts. The next step is to validate the learning potentials identified in this paper, in terms of interactions between mathematics and informatics, algorithms, and algorithmic thinking.

Acknowledgment – Work supported by the IDENTITIES Project, co-funded by the Erasmus+ Programme of the European Union under Grant Agreement n° 2019-1-IT02-KA203-063184.

References

Akkerman, S. F., & Bakker, A. (2011). Boundary crossing and boundary objects. *Review of Educational Research, 81*(2), 132–169. https://doi.org/10.3102/0034654311404435

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

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

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

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to algorithms* (Fourth Edition). The MIT Press.

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

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

Fellows, M. R., & Hoover, M. N. (1991). Perfect domination. *Australasian Journal of Combinatrics*, *3*, 141–150.

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

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

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. *2014 IEEE Integrated STEM Education Conference*, 1–4. https://doi.org/10.1109/ISECon.2014.6891029

Lindmeier, A., & Mühling, A. (2020). Keeping Secrets: K-12 Students' Understanding of Cryptography. *Proceedings of the 15th Workshop on Primary and Secondary Computing Education*. ACM, 1–10. https://doi.org/10.1145/3421590.3421630

Lodi, M., Sbaraglia, M., & Martini, S. (2022). Cryptography in Grade 10: Core Ideas with Snap! And Unplugged. *Proceedings of the 27th ACM Conference on Innovation and Technology in Computer Science Education Vol. 1*, 456–462. https://doi.org/10.1145/3502718.3524767

Modeste, S. (2016). Impact of Informatics on Mathematics and Its Teaching. In F. Gadducci & M. Tavosanis (Eds.), *History and Philosophy of Computing*. Springer, 243–255. https://doi.org/10.1007/978-3-319-47286-7_17

Skiena, S. S. (2020). *The Algorithm Design Manual*. Springer International Publishing. https://doi.org/10.1007/978-3-030-54256-6_9

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