Big Ideas of Cryptography in Primary School

Michael Lodi*
University of Bologna
Bologna, Italy
michael.lodi@unibo.it

Maria Cristina Carrisi*
University of Cagliari
Cagliari, Italy
mariacri.carrisi@unica.it

Simone Martini*
University of Bologna
Bologna, Italy
simone.martini@unibo.it

ABSTRACT

We present a learning path on cryptography for primary school students (Grade 5), which we designed and tested. The project aims to raise initial awareness of the core ideas of modern cryptography, which are fundamental concepts for becoming informed and active citizens in modern digital society. For this reason, we designed a progression of unplugged activities (sometimes integrated with task-specific, block-based programming environments) to expose students to different cryptographic techniques, where each new activity is motivated by an analysis of the criticalities encountered previously. We used unplugged activities to show that Computer Science (CS) does not necessarily imply the use of digital devices and to allow students to focus on its general scientific principles. We describe the designed learning module, discuss the main hinges in the experiment, and reflect on the lessons learned. The final evaluation showed excellent learning outcomes and high satisfaction with the activities. As cryptography uses mathematical tools (e.g., modular arithmetic, statistics), some parts of which are within reach of primary school children but rarely taught to them, it was possible to dwell on these aspects, making them experience mathematical objects in a non-standard context, and also stimulating a greater awareness of the impact of mathematics and CS in everyday life.

CCS CONCEPTS

• **Social and professional topics → K-12 education**; *Computing education programs*; • **Security and privacy → Cryptography**.

KEYWORDS

cryptography education, big ideas of cryptography, task-specific programming, unplugged, mathematics education

ACM Reference Format:
Michael Lodi, Maria Cristina Carrisi, and Simone Martini. 2024. Big Ideas of Cryptography in Primary School. In *Proceedings of the 2024 Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2024), July 8–10, 2024, Milan, Italy.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3649217.3653548

1 INTRODUCTION

The digital revolution we are experiencing significantly impacts most aspects of our lives. Education must provide students with the tools to understand a rapidly changing world. For this reason, many nations are integrating CS into the curriculum [16, 17]. Among the topics, cybersecurity is present in many curricula as early as primary school [17, p. 12], as the availability of digital communication tools exposes children to risks of which they must be aware. Although cryptography is not explicitly listed among the relevant topics for primary school, it is possible to introduce primary students to its big ideas through age-appropriate activities, as with other sciences. Moreover, cryptography is recognized by educators and students as interesting, fun, and engaging [12, 25]. CS is not (yet) a separate discipline in Italy but, according to laws, curricula, and guidelines, should be integrated into other disciplines as a transversal topic [27, 28, 31]. Cryptography can provide many links with mathematics, history, and language. Furthermore, it allows the study of aspects of CS that, instead, are explicitly listed for primary education (e.g., programming, data analysis, algorithms). We propose a learning path to teach cryptographic methods relevant to highlighting aspects of modern techniques to protect digital communication tools, which students unconsciously use daily. Given the target audience's age, we mainly designed unplugged activities (kinesthetic and fun activities to teach CS concepts without computers [6, 7]). In addition, we heavily readjusted some tools we have developed for similar courses aimed at upper secondary school students [26].

The first objective of this paper is to present the learning path we designed and describe the implemented activities. Then, we introduce the instruments and the results of our evaluation. Finally, we discuss our findings, making suggestions for improvement to enable effective adoption of the pathway.

2 BACKGROUND AND RELATED WORK

Cybersecurity is one of the fundamental concepts in the CSTA K-12 standards [14]. Regarding cryptography, they suggest starting from steganography (hiding messages inside pictures or other media) and from letter substitution methods like Caesar ciphers, moving towards modern methods, "such as public key encryption, through unplugged activities" [14, 2-NI-06].

A review of cybersecurity education in SIGCSE conferences [37] shows that only seven out of seventy papers are targeted (also) to K-12 (middle and, in particular, high school; most of the others to undergraduates). Only two of these seven contain cryptography elements, always inside more general cybersecurity activities. A review [32] on cryptography education in middle school found fifteen works, most of which present cryptography elements inside more extensive programs. Only two research groups focused on modern cryptography, not just historical ciphers [26, 33–35].

Some authors designed *unplugged* activities where students experiment with encryption and decryption algorithms, protocols, and attacks. Konak [23] proposed simple *unplugged* activities to explain classical and modern cryptosystems to middle school students with

*Also with Lab CINI "Informatica e Scuola"

ITiCSE 2024, July 8–10, 2024, Milan, Italy
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0600-4/24/07
https://doi.org/10.1145/3649217.3653548

paper, pen, scissors, boxes, and padlocks. Fees and colleagues [18] made concrete the color metaphor used for Diffie-Hellman's key exchange by having students mix food dyes to generate a shared secret key. *Hands-on* and *inquiry-based* cybersecurity/cryptography activities can improve students' self-efficacy and *problem-solving* skills [24]. Activities for older students have also been proposed (see reviews in [1, 26]).

Almost all the activities presented in the literature are one-offs. Lodi and colleagues proposed a learning path to teach high school students the big ideas of classical and modern cryptography through task-specific Snap! environments, unplugged activities, and visualizations [26].

We are unaware of any similar curriculum for primary school.

3 THE PATH

3.1 Context

The intervention was conducted in a primary school in Sardinia and involved 19 students (8 boys and 11 girls) in Grade 5 (the last year of primary school in Italy). Students had almost no experience with CS (in primary school, they only followed a couple of "Hour of Code" activities in Code.org) and had unstructured experience with digital devices based mainly on their personal use. There were 4 children with disabilities (2 of them quite severe: by law, they attend regular lessons with support teachers) and 1 with special educational needs.

Lessons were conducted during curricular hours: students participated in all activities (except for 2, due to absences). The intervention lasted 5 lessons (10 hours): 8 hours of activities and the last 2 for evaluation (satisfaction questionnaire and final assessment).

Lessons were conducted by one of the authors, a math university professor with long experience in K-12 CS teaching. The school teachers asked for 1 CS lesson, and she managed to conduct 5.

The teachers gave us an overview of students' prior knowledge of CS, math, and problem-solving skills so that we could design appropriate activities. It was ensured that children possessed the following knowledge and abilities: arithmetic operations between natural numbers and properties of the remainder of integer division, analogy between addition or subtraction and forward or backward movements in space, cyclicity of events (e.g., in measuring time), frequency of an event and its representation on histograms. In addition, some problem-solving skills (e.g., hypothesis formulation and analysis) had already been practiced.

3.2 Big ideas and relevant systems

In analogy with [26], we have chosen a progression from classical to modern cryptography systems, allowing one to experiment with more robust systems than their predecessors, thus making students experience some relevant ideas of cryptography.

After an introduction and examples of **steganography**, we started with an Italian variant of **Pig Latin** [41], usually well known by students, as a simple way to make them recognize—within a familiar context—the basic elements of a cryptosystem (key, encryption and decryption algorithms, plaintext, ciphertext). After that, we moved to the **Caesar cipher**, using it as a simple example of monoalphabetic substitution, as a way to introduce the cyclicity of modular arithmetic, a mathematical background of many cryptosystems, and as a way to show basic cryptanalysis techniques (brute-force, frequency analysis, and person-in-the-middle attacks). **Alberti cipher** was then used as a simple example of polyalphabetic substitution, useful to show how frequency attacks can fail and to introduce *One-time pad* as its extreme form. We then used **Substitution-Permutation networks** as an example of a modern symmetric block cryptosystem and to easily show avalanche effects (diffusion and confusion). Finally, we used **Diffie-Hellman** as an example of a key-generation protocol (also seminal for asymmetric cryptography) solving the key-distribution problem. Its weakness to the person-in-the-middle attack was left as an open problem.

This path allows students to encounter fundamental cryptography concepts such as the invertibility of a process and its ease or difficulty, the need to hide the relation between plaintext/key and ciphertext, the security (information-theoretic vs. computational) of a cryptographic method, and the relationship between security and the tools (technology or available information) at hand.

3.3 Teaching methodologies

Considering the participants' age and minimal experience with computers and CS, we built a mainly unplugged path. Computational tools were helpful to stimulate the need to be precise and formalize an algorithm with a limited set of unambiguous instructions. Moreover, they helped speed up generalization processes, allowing us to do repetitive tasks or to test the algorithms on longer data quickly. While maintaining an accessible register, necessary to speak with Grade 5 students, we used fundamental cryptography terminology.

3.4 Activities and tools

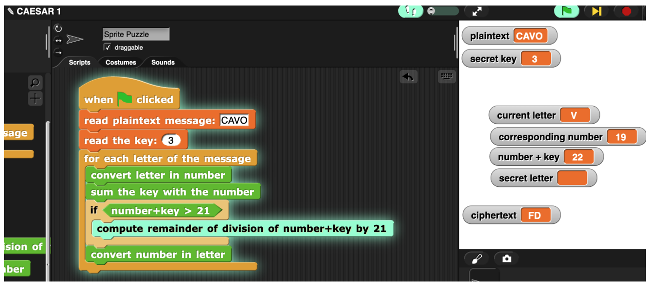
Introduction and steganography. We started with a brief introduction to CS as a science that acts, in various ways, on information. We then discussed the concept of communication with the students and the problems related to security in transmitting data. We then showed simple examples of steganography (Herodotus' wax table, servant's head shaving, vinegar writing on a boiled egg shell).

Pig Latin and Caesar cipher. As a first, simple example of encryption, we choose a variant of the "Farfallino alphabet" [40], a language game that encrypts by inserting an 'f' before each syllable of the word. To engage students, we showed a brief extract from "The Simpsons" in which the main character uses this argot. They immediately recognized the language and were able to formalize the rule. Using some variations on the argot, we introduced the idea of *security* of a cryptosystem.

The Caesar Cipher was then introduced, again focusing on identifying the key and the encryption/decryption algorithms. Various encrypting and decrypting exercises were carried out: collective, individual, and in pairs, including ones using keys other than the classical one (number 3). The pair exercise aimed to simulate real communication with all its elements. One of the teachers acted as the person-in-the-middle, either eavesdropping on the couple's key or building trust to make the parties give them the key on demand.

Students were then asked to encrypt a word where "moving to the right" in the alphabet list would overflow (e.g., the Italian word *CAVO* with key 3) to make them generalize the encryption algorithm for any key and to introduce the mathematical tool for solving the problem of unavailability of characters beyond the letter ‘Z’. Some students spontaneously found the solution of “starting over”. For the benefit of all, the following unplugged activity was used. Each student (plus teachers, to cover all 21 letters of the Italian alphabet) was given a sheet with a letter to stick to their chest; they were then arranged in a row and asked how to have letters available even in the event of an overflow. Quite quickly, the students came to position themselves in a circle. One child was asked to impersonate an arrow: after positioning himself at the center of the circle and pointing at the letter to be encrypted, he rotated 3 positions (children) to find the corresponding ciphered letter. Students were then guided in identifying, among the arithmetic operations they already knew, one that would allow them to “return to the same values”. Analyzing different options, they arrived at integer division and its remainder. They were already aware that the remainder was always smaller than the divisor, and, through examples with small numbers, they realized that it repeats cyclically. Students were suggested to match each letter with a number (starting with A=0) to apply the remainder method to the alphabet.

By this time, students were ready to formalize the encryption algorithm using mathematical operators. They collectively constructed the solution, giving instructions to the teacher, who was using, on the interactive whiteboard, a Snap! task-specific language we purposely built [11]. It provides very high-level blocks that, properly combined, allow the description of the intended algorithm (see Figure 1). The Snap! stepper also allowed a step-by-step execution of the program, monitoring the state of the machine represented in the Snap! stage (Figure 1, right). This phase was intended to show students the importance of being precise in giving instructions to a non-human executor, and to enable them to discover (or re-discover) fundamental programming constructs of selection and iteration (reflecting on the need to repeat the same operations on each letter). Moreover, using a programming environment helped test the algorithm quickly and on longer words and phrases.



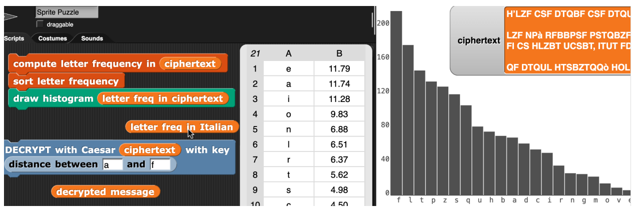
```
when [flag] clicked
read plaintext message: CAVO
read the key: 3
for each letter of the message
  convert letter in number
  sum the key with the number
  if number+key > 21
    compute remainder of division of number+key by 21
  convert number in letter
```

plaintext CAVO
secret key 3
current letter V
corresponding number 19
number + key 22
secret letter
ciphertext FD

Figure 1: Snap! task-specific programming environment

Attacks on the Caesar cipher. We introduced cryptanalysis methods. Brute force was presented via an unplugged activity inspired by [21], allowing students to reflect on *computing power* by observing that it was easy to decipher the message, but only by using many children—and therefore many attempts. It was highlighted that, given the availability of powerful computers (often with many processors), the computational time required to attack a cryptosystem is a crucial element in its security.

As for the frequency attack, we used another purpose-built Snap! environment [11] to speed up the frequency calculation, thus being able to operate on longer texts—a more realistic situation and more appropriate to see concretely the different frequency distribution of letters. We used the Snap! environment (see Fig. 2) to guide brainstorming on the interactive whiteboard. We created the histogram of letter frequencies in an assigned ciphertext and, comparing it with the corresponding histogram of letter frequency in the Italian language, students made hypotheses about the corresponding plaintext character of the most recurrent character in the ciphertext, showing a great ability in analyzing and comparing histograms. The children, together with the instructor, tested the different hypotheses until they reached a meaningful deciphered text. They observed that this attack required far fewer attempts than the brute-force one.



```
compute letter frequency in (ciphertext)
sort letter frequency
draw histogram (letter freq in ciphertext)
(letter freq in Italian)
DECRYPT with Caesar (ciphertext) with key
  distance between a and f
(decrypted message)
```

21	A	B
1	e	11.79
2	a	11.74
3	i	11.28
4	o	9.83
5	n	6.88
6	l	6.51
7	r	6.37
8	t	5.62
9	s	4.98
10	c	4.50

ciphertext
H’LZF CSF DTQBF CSF DTQU
LZF NPà RFBBPSF PSTQBZF
FI CS HLZBT UCSBT, ITUT FD
QF DTQUL HTSBZTQQò HOL

Figure 2: Snap! environment for Caesar frequency attack

SP-networks. To introduce the substitution–permutation (SP) networks, we designed an activity, inspired by the famous CS Unplugged sorting networks [13], to make students simulate one round of a simple SP-network drawn on the floor (Fig. 3).

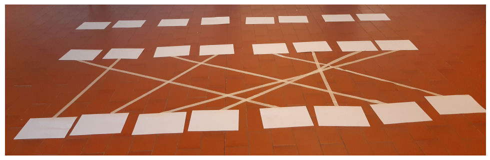


Figure 3: Unplugged Substitution-Permutation network

To represent a 4-character word, we converted each letter into a 2-digit decimal number. Therefore, we involved 8 children to represent a word, each holding a sheet with one of the 8 digits on it. Similarly, 8 other students represented the key, a 4-letter word converted to numbers. Students representing the text moved through the network from top to bottom, performing the operations they encountered, i.e., updating the digit on their sheet or their position on the network. First, they add (module 10) their own digit with the corresponding key digit (held by a child who also checked the correctness of the addition). Then, they go through the S-boxes, simulated by a child outside the network who makes a substitution of the digit based on a substitution table. Then, they followed the diagonal lines drawn on the floor to perform a permutation of the order of the digits. Finally, each 2-digit decimal number (mod 21) was converted back into letters to obtain the ciphertext. It was possible to observe that the encrypted message obtained differed significantly both from the original one and the key.

After performing this physical simulation, we moved on to using a digital tool: a spreadsheet [11] (a highly simplified version of the one we used in [26]) that simulates the network, first with one and then with multiple rounds of substitution, permutation, and key permutation, resembling the operation of modern symmetric block ciphers. This second part of the activity is not necessary to convey the main operation of an SP-network, but it is helpful in performing multiple rounds in a reasonable time and showing students that small changes in the message or the key cause significant effects on the ciphertext ("avalanche effect", confusion, and diffusion).

Diffie-Hellman key exchange. Given the students' young age, we decided to introduce the Diffie-Hellman (DH) key generation protocol using the classic metaphor of mixing colors but paying much more attention to setting up a scenario coherent with its core elements. We created (Fig. 4) a setting in which two students, whom we will call Alice and Bob (the two students at the bottom, left and right, of the picture), each had a box with 6 tempera colors, and some graduated glasses in which to dose and mix the colors. The box represents each student's private area: no one except the child operating in it could see inside. The remaining part of the table represents the public channel. The two students performed the algorithm, choosing a shared color by speaking aloud, mixing it with a secret color of their own, exchanging the mixing results via the "public table" and finally mixing the exchanged color again with the secret color. Students were very surprised to get the exact same color after the last mix, especially since the experience was repeated several times with different color choices.

In the same scenario, a third student (Fig. 4 on top) performed a person-in-the-middle to attack, by performing two key exchanges separately with Alice and with Bob, impersonating the other actor of the exchange for both.

After that, we used an animation to show the protocol's actual mathematical steps. The students had just studied exponentiation, so they were involved in calculating an actual DH key generation with very small numbers.

We also discussed the feasibility of a brute-force attack: easy with only 6 colors but very time-consuming in a realistic scenario, involving large numbers.

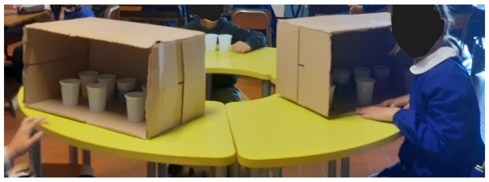


Figure 4: Diffie-Hellman key generation with tempera

Invertibility. During the experience, we stressed, on the one hand, the importance of being able to reverse the encryption procedure to recover the plaintext from the ciphertext and, on the other hand, the fact that the difficulty of reversing an algorithm when some secret information is missing (in the case of the DH activity, unmixing without knowing the original colors) is the basis of modern (asymmetric) algorithms. We carried out a series of small hands-on activities to reinforce this concept. For example, students noted that it is easy to recover a drawing made on a deflated balloon (which becomes unintelligible if you inflate it), while it is impossible to recover a drawing made on plasticine once it has been deformed. Moreover, we mixed salt with iron filings to show that a process that seems impossible to invert can be inverted simply by using a proper instrument (a magnet).

4 EXPERIENCE EVALUATION

To evaluate the cryptography course, we used two instruments. The first is a satisfaction questionnaire about liking, difficulty, and personal feelings during the activities. The second is a test to assess learning outcomes in terms of acquiring discipline-specific language and understanding the main ideas and methods of cryptography.

Both instruments were anonymous. The satisfaction questionnaire was proposed first to prevent any difficulty in solving exercises from affecting the result.

16 out of 19 students completed the questionnaire and test, as one was absent, and the two students with severe disabilities were unable to do so.

4.1 Satisfaction

We asked students about the learning experience, focusing in particular on how they felt during the activities, how much they enjoyed them, and how difficult they perceived them. We proposed a 4-point Likert scale from 'Very much' to 'Not at all', with each level also visualized by means of a smiling or frowning face.

The global satisfaction is high as all students found the activities 'Very much' or 'Rather much' interesting and easy. Moreover, most of them found the activities very useful in understanding better what CS is about and why Math is useful (Fig. 5, top chart).

Interest in specific activities is generally very high (Fig. 5, middle chart), with only slightly less interest in attacks on cryptosystems and SP-networks expressed by a few students. The same activities, together with the DH with colors (that was really appreciated, though), were also perceived as difficult by a few students (Fig. 5, bottom chart).

We also asked students to write what they learned and a comment about the activities. Most of them wrote that they learned "to encrypt and decrypt", were "sad that the lessons finished", and desired to "repeat the experience again".

4.2 Assessment

To assess different aspects, including knowledge, skills, and competencies (i.e., the integration of knowledge, skills, and dispositions to solve a problem in a new, authentic context [19]), we proposed a test with different kinds of exercises: a fill-in-the-blanks text on the most important concepts covered in the course, with the blanks in place of the 13 most relevant keywords and the 13 solutions in random order; four exercises covering the following topics: substitution cipher, transposition cipher with a permutation network, polyalphabetic cipher, and DH protocol. The English translation of the test is available at [11].

Fill-in-the-blank results are excellent, with a range of correct fillings between 9 and 13 over 13 (mean 11.8, median 12).

The subsequent exercises differed from our activities during lessons, to make students apply the knowledge acquired in new

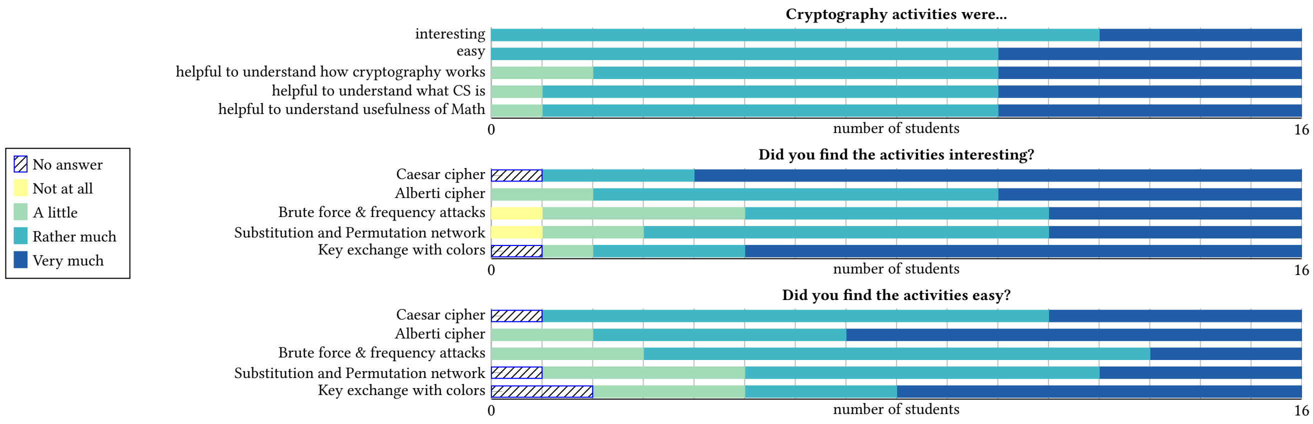


Figure 5: Results of the satisfaction questionnaire

contexts. For this reason, and also to compare with national results, two exercises come from the Bebras challenge [9, 15].

The first exercise [2] has been proposed *as is*, since it is recommended for the same age range (8-10). 81% of our students solved it correctly, compared to the 40% in Bebras “solo” and 60% in Bebras “double” of the Italian competitions [3].

Instead, the second exercise we chose [4] was recommended for the age range 12-13. We simplified the terminology in the text while retaining the same request (encrypt a word—chosen with a length of 3 instead of the original 5 letters). Only one student solved the exercise correctly and completely, compared to 80% success [5] in Bebras. We discuss this in sec. 5.

The third exercise was a simple permutation (drawn like one of the P-box in [38]), and was solved correctly by all the students.

The fourth exercise was about the color-mixing metaphor for the Diffie-Hellman key exchange. We gave students a Wikipedia picture [39], without the labels (but with a legend with the names of the colors) and asked them to pair each color with its role in the metaphor (e.g. “Alice’s secret color” or “Shared secret color”) and explain the meaning of the arrows (solution: exchange) and the plus sign (solution: color mixing). Over 8 maximum points, results show a mean score of 5.2 and a median of 5.5.

5 OBSERVATIONS AND FINDINGS

The activities generally went very well, both in terms of learning and satisfaction, showing that it is possible to integrate classical and modern cryptography activities as early as primary school, even without (at least initially) the need for digital skills from students. In the following, we detail our observations regarding specific aspects. We discuss what worked and what can be improved to help CS educators adopt and adapt our activities.

Traditional vs. innovative teaching. In general, a traditional transmissive approach is still dominant in Italian schools. This class is no exception: according to the teachers themselves, the activities we implemented were much more open-ended and kinesthetic than what the children were used to.

As they are accustomed to notional learning, it was easy for most children to remember and correctly complete the definitions in the fill-in-the-blank exercise. Conversely, during the lessons, they were displaced by—and enjoyed a little less— the more open-ended (albeit highly scaffolded) activities, as discussed in 4.1. Children showed insecurity about what to do, asking for very specific directions (e.g., where to lean, where exactly to write on the sheet, and so on).

Math and CS misconceptions. Even in these simple activities, we identified some misconceptions well-known in the literature.

Even though the results are satisfactory, it is interesting to analyze the assessment’s DH colors exercise (see 4.2). A first observation is that students confused two similar colors (*pesca* and *arancio* in our legend) and may also have been disoriented by the non-standard (like CMYK) mixing depicted in the figure [39]. More interestingly, some students grasped the meaning of the “+” operator, but in their answers, they confused the operator (i.e., an action on some object) with the result obtained (i.e., an object), stating, for example, that the plus sign stands for “the color obtained after the mixing”. This is an example of *procept*, well known in Math education [22, 36], and it refers to symbols and terms that are used for both a process and a concept, creating confusion in the students that, sometimes, cannot even recognize the presence of an operation. In CS it is also possible to find many examples of processes manipulated as objects: research on this topic is advocated [30].

Sometimes, the misconceptions had apparent effects; other times, since the introductory level of our intervention, they did not harm. For example, confinement within a well-defined domain (i.e., the Italian alphabet) in the Caesar unplugged activity necessarily led children to find a way to remain in the same domain. In this case, this confinement was functional in determining the solving strategy (“starting over”). In general, however, teachers have to pay attention to *erroneous generalization of methods that work only in a specific context* that may create misconceptions [20, 42] to recognize them and guide students towards the correct generalization, avoiding entrenched errors that can manifest much later in the learning path.

Overgeneralization misconception is also known in programming teaching as early as from the work of Du Boulay [8]. CSEd research should also focus on misconceptions in introductory activities, to avoid possible undesired effects in the future.

Interdisciplinarity. The positive interaction with the primary teacher who proposed the project was fundamental for the activity's success. During the first lesson, she provided examples and often referred to other activities already done by children in the present or previous years. For example, she cited what she had already explained about communication—allowing us to detail and generalize to non-human communication—or about the different representations of numbers (Arabic and Roman numeral systems) and other types of information (iconic and literal representation). Steganography, which we used to show other methods of protecting information, also allowed us to connect with what children are studying, for example, in the history classes (the Spartans). Moreover, the servant's head shaving has been a compelling *ante litteram* example of "time dependence" of the security of a cryptosystem.

More technically, the fact that they just studied exponentiation allowed us to show how such an important and groundbreaking cryptographic method like the Diffie-Hellman key generation worked mainly with fundamental operations they already knew (i.e., exponentiation and the remainder of integer division).

Cryptography is a field developing at the interface between CS and Math [29]. The educational opportunities and challenges of using its boundary concepts have been studied in CS education for high school teacher training [1], and should be studied even more for primary school, where discipline separation is much less rigid.

Finally, as highlighted in [10], some steganography and cryptography activities, such as Alberti's cipher disk, can be used to strengthen important language skills.

Attention to the level of difficulty. As for the pairwise activity on Caesar's cipher, we suggest directly assigning plaintext and key: in some cases, the decoding step expressed as subtraction could lead to negative results, which is beyond the knowledge of primary school students. On the other hand, this could be an opportunity in Grades 6-7: encountering this situation and the consequent need to calculate the remainder of the division between two signed numbers could motivate students toward the study of this operation, which is often overlooked and whose practical utility is often not understood.

Regarding the difficulties in the assessment exercise on the polyalphabetic cipher (see 4.2 and [4]), we note that its complete and correct resolution requires the execution of 2 fundamental tasks: 1) recognizing the position of a letter in its group of letters; 2) counting how many steps are performed to reach the proper group: a) starting the count from the first group (the original position of the arrow) for the first letter, or b) starting the count from the previous position (not from the first one). For the first letter, 50% of the students succeeded (tasks 1 and 2a). For the following letters, 69% of them succeeded in task 1 but not in task 2b, implying that the problem was related to counting how many steps to do in a circular path (sometimes it is clear they always started from the initial position instead of the previous one, other times they used an unclear approach). Some may have tried to draw specific information from the example, but this approach is incorrect, unlike the previous exercise. We underestimated the difficulty of this exercise, attributing it only to the more complex and technical language—that we simplified without considering the presence of other hidden difficulties. We also noted that this test exercise is quite different from Alberti's cipher disk one: an activity in which the key is the number of rotations instead of a word could reduce this gap.

Scientifically correct design for the unplugged activities. We strived to simplify without trivializing the fundamental concepts. We always chose metaphors that maintained consistency with the basic principles: for example, in the SP-networks letters are represented with numbers (decimal instead of binary), and the sheet containing a letter is divided into two parts to contain the two digits of the corresponding number, showing that the network works on a digital representation of information. In the DH activity, as a novel added value to previous proposals (e.g., [18]), much attention was paid to the setting, focusing on which information is public (exchanged loudly or on the table, thus audible/visible to all) and which is private (clearly demarcated with large boxes), to the role of the two sides of the communication (freely choosing their own secret color) and to the autonomy of the students involved in the experiment. We also allowed the protocol to be run several times to check its operation.

Students with disabilities. Although one of the students with severe disabilities carried out some of the activities (e.g., implementing the S-box), we suggest intense interaction with the support teachers and the continuous presence of the same teachers during all lessons to guarantee greater participation.

Use of programming. We would have liked to make the students program, having them explore the Snap! tools we built more deeply so that they could apply and reinforce (also through metacognitive reflection) the concepts they had learned. Unfortunately, this was impossible, given the almost total absence of prior programming knowledge (only a few Code.org activities were done collectively on the interactive whiteboard in previous years when the main teacher was absent) combined with limited time. In our activities with Snap!, we informally observed that students grasped fundamental programming constructs. However, we cannot speculate on the depth of understanding or learning in such a short time.

In introducing CS in school (as a discipline or, at least initially, in a cross-curricular form), it is essential to envisage a 5-year long path that includes learning programming.

Finally, we noticed some difficulty for students during unplugged activities in executing sequences of instructions longer than a few steps: learning programming could help improve this skill, too.

ACKNOWLEDGMENTS

M. Lodi's work has been supported by the Spoke 1 "FutureHPC & BigData" of the Italian Research Center on High Performance Computing, Big Data and Quantum Computing funded by MUR M4C2 Investimento 1.4: Potenziamento strutture di ricerca e creazione di campioni nazionali di R&S (M4C2-19)-Next Generation EU (NGEU).

M. C. Carrisi has been partially supported by the National Recovery and Resilience Plan (NRRP) – Project eINS Ecosystem of Innovation for Next Generation Sardinia – CUP F53C22000430001 funded by the European Union – NextGenerationEU, Code ECS0000038.

S. Martini was partially supported by: (a) SERICS (PE00000014) under the Italian MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU; (b) Italian PRIN 2020 20222BP7K3 Learning informatics; (c) INdAM GNSAGA.

REFERENCES

[1] Evmorfia-Iro Bartzia, Michael Lodi, Marco Sbaraglia, Simon Modeste, Viviane Durand-Guerrier, and Simone Martini. 2024. An Unplugged Didactical Situation on Cryptography between Informatics and Mathematics. *Informatics in Education* 23, 1 (2024), 25–56. https://doi.org/10.15388/infedu.2024.06

[2] Bebras. 2021. Bebras 2021-HR-03 *Lettere*. https://bebras.it/platform/html/player_teacher.html?class=screenshot&code=2021_Q01

[3] Bebras. 2021. Statistiche gare Bebras italiano 2021. https://bebras.it/2122/Statistiche+gare.html

[4] Bebras. 2022. Bebras 2022-SK-04 *Crittografia*. https://bebras.it/platform/html/player_teacher.html?class=screenshot&code=2022_Q14

[5] Bebras. 2022. Statistiche gare Bebras italiano 2022. https://bebras.it/2223/Statistiche+gare.html

[6] Tim Bell, Jason Alexander, Isaac Freeman, and Mick Grimley. 2009. Computer Science Unplugged: school students doing real computing without computers. *New Zealand Journal of Applied Computing and Information Technology* 13, 1 (2009), 20–29.

[7] Tim Bell and Michael Lodi. 2019. Constructing Computational Thinking Without Using Computers. *Constructivist Foundations* 14, 3 (2019), 342–351. https://constructivist.info/14/3/342.bell

[8] Benedict Du Boulay. 1986. Some Difficulties of Learning to Program. *Journal of Educational Computing Research* 2, 1 (1986), 57–73. https://doi.org/10.2190/3LFX-9RRF-67T8-UVK9

[9] Annalisa Calcagni, Violetta Lonati, Dario Malchiodi, Mattia Monga, and Anna Morpurgo. 2017. Promoting Computational Thinking Skills: Would You Use this Bebras Task?. In *Informatics in Schools: Focus on Learning Programming (ISSEP 2017, Vol. 10696)*, Valentina Dagienė and Arto Hellas (Eds.). Springer International Publishing, Cham, 102–113. https://doi.org/10.1007/978-3-319-71483-7_9

[10] Maria Cristina Carrisi. 2023. Crittografia e steganografia alla scuola primaria: un'attività di coding ricca di potenzialità. In *[ITADINFO] Convegno Italiano sulla Didattica dell'Informatica* (Bari, Italy) *(ITADINFO 2023)*, Enrica Gentile and Mattia Monga (Eds.). Università degli Studi di Bari Aldo Moro, Bari, Italy, 24–29. https://www.itadinfo.it/attidelconvegno/

[11] Maria Cristina Carrisi, Michael Lodi, and Simone Martini. 2024. Cryptography Teaching Materials for Primary School. https://bigideascryptok12.bitbucket.io/primary

[12] EdWeek Research Center. 2020. *The State of Cybersecurity Education in K-12 Schools: Results of a National Survey.* Technical Report. cyber.org. https://cyber.org/sites/default/files/2020-06/The%20State%20of%20Cybersecurity%20Education%20in%20K-12%20Schools.pdf

[13] CS Unplugged. [n. d.]. Sorting networks. Retrieved January 15, 2024 from https://www.csunplugged.org/en/topics/sorting-networks/

[14] CSTA. 2017. *CSTA K-12 Computer Science Standards, rev. 2017.* Technical Report. Computer Science Teachers Association. http://www.csteachers.org/standards

[15] Valentina Dagienė and Gerald Futschek. 2008. Bebras International Contest on Informatics and Computer Literacy: Criteria for Good Tasks. In *Informatics Education - Supporting Computational Thinking*, Roland T. Mittermeir and Maciej M. Sysło (Eds.). Springer, Berlin, Heidelberg, 19–30.

[16] Department of Education. 2013. *National curriculum in England: computing programmes of study.* https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study

[17] European Commission / EACEA / Eurydice. 2022. *Informatics education at school in Europe.* Publications Office of the European Union, Luxembourg. https://eurydice.eacea.ec.europa.eu/publications/informatics-education-school-europe

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

[19] Stephen Frezza, Mats Daniels, Arnold Pears, Åsa Cajander, Viggo Kann, Amanpreet Kapoor, Roger McDermott, Anne-Kathrin Peters, Mihaela Sabin, and Charles Wallace. 2018. Modelling competencies for computing education beyond 2020: a research based approach to defining competencies in the computing disciplines. In *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education* (Larnaca, Cyprus) *(ITiCSE 2018 Companion)*. Association for Computing Machinery, New York, NY, USA, 148–174. https://doi.org/10.1145/3293881.3295782

[20] Toshiakira Fujii. 2020. *Misconceptions and Alternative Conceptions in Mathematics Education.* Springer International Publishing, Cham, 625–627. https://doi.org/10.1007/978-3-030-15789-0_114

[21] Christian Garro. 2024. BRUTE FORCE DISTRIBUITO. In *InformaticaSenzaPC*, Sara Capecchi (Ed.). Università degli Studi di Torino, Turin, Italy. Retrieved April 13, 2024 from https://informaticasenzapc.di.unito.it/brute-force-distribuito/

[22] Eddie Gray and David Tall. 1992. Success and Failure in Mathematics: The Flexible Meaning of Symbols as Process and Concept. *Mathematics Teaching* 142 (1992), 6–10.

[23] Abdullah Konak. 2014. A cyber security discovery program: Hands-on cryptography. In *2014 IEEE Integrated STEM Education Conference*. IEEE, 1–4. https://doi.org/10.1109/ISECon.2014.6891029

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

[25] Anke Lindmeier and Andreas Mühling. 2020. Keeping Secrets: K-12 Students' Understanding of Cryptography. In *Proceedings of the 15th Workshop on Primary and Secondary Computing Education* (Virtual Event, Germany) *(WiPSCE '20)*. ACM, New York, NY, USA, Article 14, 10 pages. https://doi.org/10.1145/3421590.3421630

[26] Michael Lodi, Marco Sbaraglia, and Simone Martini. 2022. Cryptography in Grade 10: Core Ideas with Snap! and Unplugged. In *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 1* (Dublin, Ireland) *(ITiCSE '22)*. Association for Computing Machinery, New York, NY, USA, 456–462. https://doi.org/10.1145/3502718.3524767

[27] MIM. 2023. LINEE GUIDA PER LE DISCIPLINE STEM. https://www.miur.gov.it/documents/20182/0/Linee+guida+STEM.pdf/2aa0b11f-7609-66ac-3fd8-2c6a03c80f77?version=1.0&t=1698173043586

[28] MIUR. 2017. INDICAZIONI NAZIONALI E NUOVI SCENARI. https://www.miur.gov.it/documents/20182/0/Indicazioni+nazionali+e+nuovi+scenari/

[29] Simon Modeste. 2016. Impact of Informatics on Mathematics and Its Teaching. In *History and Philosophy of Computing*, Fabio Gadducci and Mirko Tavosanis (Eds.). Springer International Publishing, Cham, 243–255.

[30] Reinhard Oldenburg. 2011. Reification and symbolization. In *Proceedings of the 11th Koli Calling International Conference on Computing Education Research* (Koli, Finland) *(Koli Calling '11)*. Association for Computing Machinery, New York, NY, USA, 49–53. https://doi.org/10.1145/2094131.2094141

[31] Italian Parliament. 2021. Italian Law 233, Dec. 29, 2021. https://www.gazzettaufficiale.it/eli/gu/2021/12/31/310/so/48/sg/pdf

[32] Nathan Percival, Sashank Narain, and Claire Seungeun Lee. 2023. Modern Cryptography Education of Middle School Students: A Review of Current Works. In *Proceedings of the 24th Annual Conference on Information Technology Education* (Marietta, GA, USA) *(SIGITE '23)*. Association for Computing Machinery, New York, NY, USA, 33–38. https://doi.org/10.1145/3585059.3611428

[33] Nathan Percival, Pranathi Rayavaram, Sashank Narain, and Claire Seungeun Lee. 2022. CryptoScratch: Developing and evaluating a block-based programming tool for teaching K-12 cryptography education using Scratch. In *2022 IEEE Global Engineering Education Conference (EDUCON)*. 1004–1013. https://doi.org/10.1109/EDUCON52537.2022.9766637

[34] Nathan Percival, Pranathi Rayavaram, Sashank Narain, and Claire Seungeun Lee. 2022. CryptoScratch: Teaching Cryptography with Block-based Coding. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 2* (Providence, RI, USA) *(SIGCSE 2022)*. Association for Computing Machinery, New York, NY, USA, 1087. https://doi.org/10.1145/3478432.3499067

[35] Pranathi Rayavaram, Ashwin Jagadeesha, Sashank Narain, and Claire Seungeun Lee. 2023. Designing a Visual Cryptography Curriculum for K-12 Education. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 2* (Toronto ON, Canada) *(SIGCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 1319. https://doi.org/10.1145/3545947.3576266

[36] David Tall, Michael Thomas, Gary Davis, Eddie Gray, and Adrian Simpson. 1999. What Is the Object of the Encapsulation of a Process? *The Journal of Mathematical Behavior* 18, 2 (1999), 223–241. https://doi.org/10.1016/S0732-3123(99)00029-2

[37] Valdemar Švábenský, Jan Vykopal, and Pavel Čeleda. 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* (Portland, OR, USA) *(SIGCSE '20)*. ACM, New York, NY, USA, 2–8. https://doi.org/10.1145/3328778.3366816

[38] Wikimedia Commons. 2021. *File:SubstitutionPermutationNetwork2.png — Wikimedia Commons, the free media repository.* Retrieved January 16, 2024 from https://commons.wikimedia.org/w/index.php?title=File:SubstitutionPermutationNetwork2.png&oldid=570824875

[39] Wikimedia Commons. 2023. *File:Diffie-Hellman Key Exchange.svg — Wikimedia Commons, the free media repository.* Retrieved January 16, 2024 from https://commons.wikimedia.org/w/index.php?title=File:Diffie-Hellman_Key_Exchange.svg&oldid=736190254

[40] Wikipedia contributors. 2023. *Farfallino alphabet — Wikipedia, The Free Encyclopedia.* Retrieved January 11, 2024 from https://en.wikipedia.org/w/index.php?title=Farfallino_alphabet&oldid=1181499638

[41] Wikipedia contributors. 2024. *Pig Latin — Wikipedia, The Free Encyclopedia.* Retrieved January 11, 2024 from https://en.wikipedia.org/w/index.php?title=Pig_Latin&oldid=1193801075

[42] Rosetta Zan. 2007. *Difficoltà in matematica.* Springer-Verlag, Milan, Italy.