Rendiconti Sem. Mat. Univ. Pol. Torino
Vol. 80, 2 (2022), 49 – 81

M. Lodi - M. Sbaraglia - S. Martini

PROGRAMMARE PER IMPARARE LA CRITTOGRAFIA AL LICEO MATEMATICO

Abstract. Descriviamo un corso di introduzione alle "grandi idee" della crittografia, pensato per il secondo anno del Liceo Matematico. Sua caratteristica fondamentale è l'approccio "per scoperta": viene proposta una successione di crittosistemi (dai classici ai più moderni), di ciascuno dei quali lo studente puó sperimentare caratteristiche, possibili attacchi e limiti, sentendo la necessità di scoprire il successivo. Abbiamo usato Snap! (un linguaggio di programmazione visuale a blocchi) sia per costruire dei *playground* (linguaggi di programmazione *task-specific*, con una ridottissima selezione mirata di istruzioni) con cui sperimentare i diversi schemi, sia per guidare un'attività *unplugged* sul protocollo di Diffie-Hellman. Il lavoro presenta in dettaglio il percorso, le attività e il materiale, nonché una prima valutazione dell'intervento didattico, condotta dopo due edizioni (la prima online e la seconda in presenza).

1. Introduzione

La crittografia è un ingrediente essenziale di moltissime attività e strumenti della nostra società contemporanea, digitale e connessa. Messaggistica istantanea e *social network*, acquisti online di beni e servizi, *trading online*, ma anche identità digitale ed *e-governance*, sono solo alcuni degli scenari contemporanei che sono possibili solo grazie alla crittografia moderna. Vari quadri di riferimento europei (come DigComp [6]) e curricoli internazionali (quali gli standard americani [18] e il curricolo informatico inglese [9]) includono competenze relative alla sicurezza informatica (*cybersecurity*). Alcune di queste competenze sono più orientate all'utilizzo della sicurezza informatica per scopi precisi (personali o aziendali), altre invece riguardano la comprensione dei suoi principi fondanti. In ogni caso, questi documenti riconoscono che le competenze legate alla sicurezza informatica sono essenziali affinché gli studenti possano diventare cittadini attivi della società digitale.

La crittografia è uno dei pilastri della sicurezza informatica. Per di più, gli studenti novizi riconoscono nella crittografia un contesto interessante per le lezioni di informatica [23, p. 3]. L'istruzione pre-universitaria non ha l'obiettivo di formare professionisti, ma si propone di aiutare gli studenti a comprendere il mondo in cui viviamo per potervi prendere parte attivamente, perseguendo i propri obiettivi. Riteniamo quindi che sia importante per gli studenti conoscere e comprendere i principi della crittografia e la loro importanza per le attività e gli strumenti della società digitale.

Sulla base di questi presupposti, abbiamo concepito un breve corso di crittografia. Si tratta di un corso senza prerequisiti specifici, costruito da un lato attorno ai concetti fondamentali della crittografia, e dall'altro fortemente basato su diversi tipi di attività pratiche che permettono di fare esperienza diretta e attiva di quei concetti. Questa impostazione trova le sue radici nella ricerca educativa, che ha

ampiamente documentato l'efficacia delle metodologie attive e cooperative nel favorire l'apprendimento [26, p. 304]. Abbiamo quindi ideato e sviluppato attività innovative, pratiche, interattive, che fossero il più possibile immediatamente significative per giovani studenti alle prime armi. In particolare abbiamo costruito dei *playground* crittografici, cioè degli ambienti digitali di sperimentazione in cui gli studenti, tramite la programmazione visuale a blocchi, potessero usare, comprendere e attaccare alcuni tra i più significativi cifrari classici (ad esempio, il cifrario di Cesare e il cifrario *One-time pad*). Abbiamo inoltre sviluppato un'attività *unplugged* (si tratta di attività che insegnano concetti informatici – solitamente algoritmi notevoli – senza l'uso di elaboratori ma con giochi fisici e materiali comuni [2]) ma remota (la prima iterazione del corso, nel febbraio 2021, si è tenuta a distanza a causa della pandemia di COVID-19) in modo che gli studenti potessero eseguire, a coppie, il protocollo di Diffie-Hellman per generare (in modo sicuro, ma comunicando solo su un canale insicuro) una chiave di comunicazione segreta. Abbiamo realizzato entrambe le tipologie di attività usando Snap!, un linguaggio di programmazione visuale a blocchi, concepito per essere usato facilmente anche dai novizi di programmazione. Come detto, la prima iterazione del corso (febbraio 2021 per l'a.s. 2020/21) si è tenuta esclusivamente a distanza a causa della pandemia, mentre è stato possibile organizzare una seconda iterazione del corso in presenza (novembre-dicembre 2021 per l'a.s. 2021/22).

Il percorso didattico include alcuni tra gli schemi e i sistemi crittografici più significativi (classici e moderni), selezionati affinché siano rappresentanti emblematici di alcune idee chiave della crittografia. Al fine di realizzare una progressione motivante, l'introduzione di un nuovo schema o sistema è sempre innescata dalla necessità (che cerchiamo di stimolare negli studenti – in linea con una metodologia didattica da noi altrove proposta: il *Necessity Learning Design* [32]) di superare i limiti del sistema (o dei sistemi) precedente.

Questo articolo presenta il percorso che abbiamo progettato per introdurre le idee e principi fondamentali della crittografia (sezione 2). Discuteremo nel dettaglio i concetti e gli schemi che di volta in volta guidano la progressione didattica (2.3) e descriveremo lo sviluppo e la sperimentazione delle attività didattiche realizzate con Snap! (sia i *playground* crittografici sia l'attività *unplugged* di Diffie-Hellman, 2.5). Presenteremo i risultati della raccolta dati su due iterazioni del nostro percorso (sezioni 4 e 5) sia dal punto di vista della comprensione dei concetti crittografici affrontati (4.1), sia per quel che riguarda la soddisfazione e l'utilità percepite dagli studenti stessi al termine di ciascuna iterazione (4.2). Discuteremo infine che cosa ha funzionato e cosa puó essere migliorato (sezione 5.1) e come aiutare i docenti di Informatica (ma anche di Matematica, Educazione tecnica, e in generale di discipline STEM) ad adottare (e adattare) il nostro percorso didattico e le sue attività pratiche in diversi contesti (5.3).

Questo articolo è una versione estesa – che approfondisce la progettazione del percorso e analizza anche la seconda iterazione dell'attività didattica – di [25].

2. Il nostro corso

2.1. Contesto: il *Liceo Matematico*

In Italia, i Licei "forniscono allo studente gli strumenti culturali e metodologici per una comprensione approfondita della realtà, affinché egli si ponga, con atteggiamento razionale, creativo, progettuale e critico, di fronte alle situazioni, ai fenomeni e ai problemi, ed acquisisca conoscenze, abilità e competenze coerenti con le capacità e le scelte personali e adeguate al proseguimento degli studi di ordine superiore, all'inserimento nella vita sociale e nel mondo del lavoro" [29].

Il *Liceo Matematico* è un progetto didattico sperimentale a livello nazionale, che coinvolge oggi più di 140 scuole. Si tratta di attività extra-curricolari, disponibili per gli studenti (che partecipano su base volontaria) di tutti i tipi di liceo. Gli studenti sono coinvolti in attività di natura laboratoriale, affinché possano fare esperienze interdisciplinari che coinvolgano Matematica e altre discipline. L'obiettivo del *Liceo Matematico* non consiste tanto nel presentare agli studenti nuove nozioni, ma nel cercare di farli riflettere in maniera interdisciplinare sui fondamenti dei saperi, in modo che essi possano ampliare i loro strumenti di lettura della realtà e in ultima analisi allargare i loro orizzonti culturali [22].

Il nostro breve corso di crittografia è stato concepito e sviluppato nel contesto di un Liceo Matematico e, in linea con gli obiettivi generali di questa sperimentazione nazionale, non punta a trasmettere nuove nozioni (tecniche e professionalizzanti) ma a far comprendere agli studenti l'importanza della crittografia oggi, facendo sí che ne sperimentino concretamente i principi fondamentali e le idee trasformative, attraverso un approccio didattico intrinsecamente interdisciplinare.

2.2. Due iterazioni: online e in presenza

Abbiamo svolto due iterazioni del corso con gli studenti che hanno scelto il percorso extra-curricolare del *Liceo Matematico* in un Liceo di Casalecchio di Reno (Bologna).

La prima iterazione si è svolta con gli studenti di II liceo dell'anno scolastico 2020/21 e la seconda con gli studenti di II liceo dell'a.s. 2021/22.

Il corso è stato tenuto da due degli autori di questo articolo, che sono ricercatori in Didattica dell'Informatica e hanno esperienza nell'insegnamento dell'Informatica nella scuola secondaria di secondo grado.

Prima iterazione (online)
La prima iterazione si è svolta in Febbraio 2021.

Abbiamo tenuto quattro lezioni (da 2 ore ciascuna) di martedí pomeriggio. A causa della pandemia di COVID-19, le lezioni si sono tenute online, attraverso la piattaforma Google Meet adottata dalla scuola.

Hanno partecipato 15 studenti (5 ragazze e 10 ragazzi).

Nessuno di loro aveva precedenti esperienze di programmazione.

Seconda iterazione (in presenza)

La seconda iterazione si è svolta tra Novembre e Dicembre 2021, in 5 lezioni (da 2 ore ciascuna) di martedí pomeriggio.

Le lezioni si sono svolte, tutte in presenza, nel laboratorio informatico della scuola, in cui era disponibile un computer per ogni studente.

Hanno partecipato 13 studenti (5 ragazze e 8 ragazzi).

Tre studenti avevano già svolto poche semplici attività di programmazione visuale a blocchi alla scuola primaria o secondaria di primo grado. Tutti gli altri non avevano nessuna esperienza pregressa di programmazione.

2.3. Un percorso guidato dai limiti dei crittosistemi precedenti

Il corso si propone di insegnare le idee fondamentali della crittografia facendo sperimentare in prima persona agli studenti schemi e protocolli crittografici significativi, da quelli classici a quelli più moderni. Gli studenti, usando la programmazione a blocchi, sperimentano meccanismi di cifratura, decifratura e di attacco di alcuni crittosistemi e, con una simulazione interattiva a coppie, generano una chiave segreta condivisa. Tali attività pratiche hanno l'intento di aiutare gli studenti a comprendere i principi crittografici degli schemi e dei protocolli incontrati, anche sperimentandone i limiti. In particolare, gli studenti incontrano schemi e protocolli in una successione progettata in modo che l'introduzione di un nuovo sistema sia sempre motivata dalla necessità di superare i limiti del o dei precedenti, progredendo sempre verso sistemi più sicuri. Il corso affronta anche alcuni concetti matematici fondamentali per la crittografia (ad esempio, l'aritmetica modulare), cercando di mostrare, con un approccio interdisciplinare, il loro ruolo nel concepimento e nella realizzazione dei sistemi incontrati.

Idee e principi della crittografia attraverso sistemi significativi

Come detto, la scelta dei crittosistemi è stata dettata dalle idee e dai principi della crittografia che è possibile insegnare attraverso di essi.

Il cifrario di Cesare (presentato nella versione in cui la chiave è un intero positivo che indica lo scostamento nell'alfabeto) è stato scelto perché ha un funzionamento semplice e quindi risulta comodo usarlo per introdurre in un contesto concreto i concetti di base quali chiave, messaggio in chiaro, messaggio cifrato, algoritmo di cifratura e decifratura. Inoltre, è facile eseguire attacchi a forza bruta e basati sulle frequenze, nonché ragionare sulle motivazioni della sua vulnerabilità a questi due attacchi (rispettivamente: poche chiavi e il fatto che lo stesso carattere in chiaro venga sempre mappato nello stesso carattere cifrato).

In contrasto con i percorsi tradizionali, che prendono in esame molti cifrari storici, abbiamo deciso di fare solo qualche accenno al cifrario di Vigenère e a Enigma. Per favorire una comprensione più diretta, i cifrari polialfabetici vengono presentati come cifrari in cui la chiave è composta da una sequenza finita (di lunghezza fissata) di "chiavi di Cesare": la prima lettera del messaggio verrà cifrata con Cesare usando il primo numero della sequenza-chiave, la seconda ancora Cesare ma con il secondo numero della sequenza-chiave e così via. Pur seguendo questo semplice approccio, è

comunque possibile mostrare facilmente le vulnerabilità dei cifrari polialfabetici (ad esempio, parole molto comuni che si ripetono a distanza multipla della lunghezza della chiave permettono di scoprire tale distanza e quindi attaccare con le frequenze).

Proprio questa vulnerabilità suggerisce di passare a un approccio in cui la chiave è una sequenza di numeri casuali (che continuiamo a chiamare informalmente "chiavi di Cesare" sempre per favorire la comprensione) lunga quanto il messaggio: ogni lettera del messaggio in chiaro verrà cifrata con un diverso numero appartenente alla chiave. Viene in questo modo introdotto il cifrario *One-time pad*. Attraverso la relativa attività pratica, gli studenti si accorgono ben presto che, per messaggi lunghi, ogni lettera ha la stessa probabilità di essere cifrata con tutte le possibili "chiavi di Cesare", e questo "appiattisce" le frequenze di ciascun carattere nel messaggio cifrato, vanificando quindi gli attacchi basati sulle frequenze. Attaccare a forza bruta *One-time pad* è ancora più sorprendente: visto che le chiavi sono tutte le possibili sequenze di numeri (da 1 alla lunghezza dell'alfabeto), il messaggio cifrato ha la stessa probabilità di essere stato originato da ogni messaggio in chiaro. Un attacco a forza bruta produce tutte le disposizioni con ripetizione sull'alfabeto lunghe quanto il messaggio, ossia include tutte le possibili sequenze di caratteri di una data lunghezza su quell'alfabeto, senza svelare nessuna informazione su quale tra esse possa essere il messaggio in chiaro. Si suggerisce quindi l'idea che *One-time pad* non rivela nessuna informazione sul messaggio originale e sulla chiave (a patto che venga cambiata ogni volta e sia veramente casuale), realizzando quindi la sicurezza perfetta. Sicurezza perfetta ma concretamente insostenibile, vista la difficoltà di generare, usare e scambiare chiavi davvero casuali, ogni volta diverse, lunghe quanto il messaggio in chiaro.

Si fa qualche accenno ai cifrari a trasposizione: il meccanismo della trasposizione e quello della sostituzione (ampiamente sperimentato con Cesare e *One-time pad*) sono elementi fondamentali dei cifrari moderni.

Si introducono quindi i moderni cifrari a blocchi. È giunto il momento di vedere come gli elaboratori elettronici (senza i quali la crittografia moderna non potrebbe concretamente realizzarsi) rappresentano l'informazione e in particolare i caratteri, ossia con schemi di codifica che, in ultima analisi, rappresentano tutto con sequenze di soli 0 e 1. Si mostrano anche semplici operazioni (che i calcolatori eseguono molto velocemente) su queste cifre binarie (bit, *binary digits*). In particolare si illustra lo XOR (disgiunzione esclusiva, che vale 1 solo se esattamente uno tra i suoi due operandi è 1), molto usato in crittografia dato che nasconde ogni informazione sugli operandi e di conseguenza sul messaggio in chiaro. Per ogni bit cifrato, infatti, sappiamo solo se il bit in chiaro era uguale o diverso al corrispondente bit della chiave (che è segreta per il principio di Kerckhoffs). Il passaggio ai bit è necessario per mostrare importanti proprietà dei cifrari a blocchi. Ad esempio, il fatto che opportune e ripetute sostituzioni e permutazioni di bit generino il cosiddetto "effetto valanga" (un piccolo cambiamento nel testo in chiaro e/o nella chiave provoca un grande cambiamento nel testo cifrato), garantendo così le proprietà di confusione e diffusione. Pur limitandosi ad analizzare con reti a sostituzione e permutazione semplificate, si possono comunque comprendere i principi di funzionamento dei moderni sistemi simmetrici quali AES. Questo contesto offre la possibilità di discutere del modello di sicurezza: non più perfetta, come quella

di *One-time pad*, ma computazionale, cioè basata sul fatto che tali cifrari non siano attaccabili in tempi umanamente utili (con la potenza di calcolo odierna e degli immediati anni a venire). I moderni cifrari a blocchi basano comunque la loro sicurezza sull'esistenza di una chiave segreta e condivisa tra le parti; in altre parole, non risolvono il problema della distribuzione delle chiavi.

Si presenta finalmente il protocollo di scambio di chiave di Diffie-Hellman, che mostra come, sorprendentemente, sia possibile generare un segreto condiviso utilizzando un canale di comunicazione insicuro. Si tratta di un protocollo semplice da comprendere (se si tralasciano i dettagli sui generatori modulari, la matematica su cui si basa è comprensibile a studenti liceali: aritmetica modulare e numeri primi) ed è tuttora ampiamente usato in pratica. Ha inoltre un'importanza storica fondamentale, dagli enormi risvolti sociali, visto che ha posto le basi per la crittografia asimmetrica. È già possibile incontrare in tale protocollo l'idea di funzione unidirezionale: facile da calcolare ma difficile da invertire. Siamo nuovamente in uno scenario di sicurezza computazionale: la sicurezza non è perfetta, matematicamente dimostrata, ma si basa sul fatto che, almeno per il momento, non siamo stati in grado di sviluppare un algoritmo con cui un elaboratore possa calcolare in modo efficiente il logaritmo discreto. Lo scambio di chiave di Diffie-Hellman non è tuttavia perfetto, infatti nel contesto del suo utilizzo è facile far comprendere agli studenti come funziona un attacco *person-in-the-middle*: una terza parte si frappone tra due attori che intendono generare un segreto condiviso, ed esegue il protocollo con ciascuna delle due parti, "impersonando" l'altra parte in entrambe le comunicazioni.

A questo punto, sfruttiamo tale rilevante limite del protocollo di Diffie-Hellman per introdurre la necessità di sistemi che garantiscano non solo la segretezza di una comunicazione ma anche la sua autenticità. Ancora una volta si cerca di introdurre un fondamentale concetto crittografico (autenticazione) nel contesto concreto dell'uso di un sistema (opportunamente scelto) per facilitare la comprensione degli studenti. Vista la complessità dei moderni sistemi asimmetrici, questi si introducono solo in modo generale e intuitivo, tramite la metafora delle due chiavi diverse (pubblica e privata) ma legate: ció che è stato chiuso con una puó essere aperto solo con l'altra e viceversa. Questo permette di spiegare ad alto livello come usarle sia per l'autenticazione ("chiudere" un messaggio con la chiave privata del mittente) sia per la segretezza ("chiudere" un messaggio con la chiave pubblica del destinatario). Si puó far intuire agli studenti come funzioni matematicamente il legame tra chiave privata e chiave pubblica, evidenziando la chiara disparità di potenza di calcolo necessaria per moltiplicare due primi molto grandi (quasi istantanea usando linguaggi di programmazione come Python) e quella per fattorizzare il loro prodotto (che puó richiedere molti mesi di calcolo in potenti supercomputer). Questa idea è alla base di uno dei più noti ed usati sistemi moderni a chiave asimmetrica: RSA. Anche questi sistemi non sono perfetti: richiedono infatti molta potenza di calcolo (diversamente dai moderni cifrari a blocchi, computazionalmente molto efficienti).

Anche alla luce di quest'ultima considerazione, il percorso si conclude cercando di mettere insieme i diversi pezzi del puzzle visti fino a quel momento. Si mostra come, ad esempio, nei moderni sistemi di messaggistica istantanea (come WhatsApp o

Telegram):

- si usa un sistema asimmetrico per essere sicuri dell'identità delle due parti che si scambiano messaggi (autenticazione);
- si esegue il protocollo di Diffie-Hellman per generare in modo sicuro una chiave segreta condivisa;
- si usa poi quella chiave segreta condivisa per comunicare segretamente tramite sistemi simmetrici, computazionalmente veloci ed efficienti.

Il percorso proposto si ferma qui, ma potrebbe essere ulteriormente ampliato, ad esempio sempre a partire dalle criticità (come quelle sollevate dalle potenzialità della computazione quantistica) dei sistemi studiati e mostrarne le possibili soluzioni.

Il percorso è riassunto nello schema seguente, che per ciascun sistema riporta: perché è stato scelto come rappresentante della classe a cui appartiene; le motivazioni didattiche di tale scelta; le limitazioni che gli studenti incontrano e che supportano la transizione al sistema successivo.

Cifrario di Cesare

- *Rappresentativo per:* cifrari a sostituzione monoalfabetica
- *Motivazioni:* esempio di base di crittografia a chiave simmetrica; facile mostrare gli elementi tipici del crittosistema (testo in chiaro, testo cifrato, funzioni di cifratura/decifratura, chiave) e semplici attacchi su di esso; facile da capire e facile sperimentare con esso

↓ *Problemi da superare:* attaccabile sia con forza bruta che con analisi di frequenza

Cifrario *One-time pad*

- *Rappresentativo per:* cifrari polialfabetici (portati all'estremo); sicurezza perfetta; resistente sia agli attacchi brute-force che a quelli con frequenze (nessun indizio sulla chiave o sul testo in chiaro a partire dal testo cifrato)
- *Motivazioni:* facile da introdurre come "un Cesare diverso per ogni lettera"

↓ *Problemi da superare:* problema di distribuzione delle chiavi; problemi di fattibilità (e.g., chiavi lunghe quanto il messaggio, casuali e nuove per ogni messaggio)

Semplice rete a sostituzione-permutazione

- *Rappresentativo per:* moderni crittosistemi simmetrici a blocchi (ad esempio, AES); confusione e diffusione (effetto valanga); implementazione efficiente; "solo" computazionalmente sicuro

- *Motivazioni:* introdurre la manipolazione di bit; capire come i moderni sistemi di crittografia sono implementati con i computer

↓ *Problemi da superare:* il problema della distribuzione delle chiavi

Protocollo di scambio di chiave di Diffie-Hellman

- *Rappresentativo per:* protocolli di generazione di chiave condivisa; soluzione innovativa al problema della distribuzione delle chiavi
- *Motivazioni:* capire come il logaritmo discreto (facile da calcolare, difficile da invertire) permette di generare un segreto condiviso su un canale pubblico (insicuro)

↓ *Problemi da superare:* attacco *person-in-the-middle*

Idea di segretezza e autenticazione a chiave pubblica

- *Rappresentativo per:* crittosistemi asimmetrici
- *Motivazioni:* comprendere che le proprietà di alcune funzioni matematiche (ad esempio, la fattorizzazione dei primi) possono essere utilizzate per ottenere sia la segretezza che l'autenticazione

↓ *Problemi da superare:* computazionalmente costosi

Idea dei crittosistemi ibridi

- *Rappresentativo per:* i complessi sistemi crittografici moderni
- *Motivazioni:* imparare che il meglio dei crittosistemi simmetrici e asimmetrici vengono combinati in pratica oggi; capire come funzionano molti servizi moderni (ad esempio, la messaggistica istantanea crittografata *end-to-end*)

○ *Problemi da superare: non sollevati nel corso*

2.4. Contenuti delle lezioni

Abbiamo istanziato il percorso progettato in 4 lezioni nella prima iterazione e in 5 nella seconda. Per entrambe le iterazioni, seguono i contenuti riportati in modo schematico.

Prima iterazione

Lezione 1

- Il dibattito sociale sulla crittografia nella comunicazione digitale
- Cifrario di Cesare: cifratura, decifratura, attacco a forza bruta
- Compiti a casa: trasposizione versus sostituzione, principio di Kerckhoffs

Lezione 2

- Cifrario di Cesare: attacco con frequenze
- *One-time pad*: cifratura, decifratura, attacco con frequenze
- Compiti a casa: codifica dei caratteri come bit, esempio di crittosistema didattico che usa XOR e permutazione di bit, accenni ai moderni cifrari a blocchi (DES, AES)

Lezione 3

- *One-time pad*: attacco a forza bruta, sicurezza perfetta, limiti, problema della distribuzione delle chiavi
- Protocollo di Diffie-Hellman: simulazione con i colori
- Compiti a casa: quiz su *One-time pad* e sul protocollo di Diffie-Hellman

Lezione 4

- Matematica del protocollo di Diffie-Hellman: aritmetica modulare, esponenziale e sua inversione, primi e coprimi (e accenni ai generatori)
- Protocollo di Diffie-Hellman: esempio con numeri, sicurezza computazionale, attacco *person-in-the-middle*
- Crittografia asimmetrica: terminologia, proprietà delle coppie di chiavi pubblica/privata, schemi (non tecnici, ad alto livello) di autenticazione e segretezza, idea intuitiva di funzione unidirezionale (moltiplicazione di primi versus fattorizzazione)
- Unire il tutto: come combinare intuitivamente schemi asimmetrici e simmetrici per l'autenticazione e la segretezza
- Compiti a casa: questionario di soddisfazione, valutazione tramite completamento di riassunto

Seconda iterazione
Nella seconda iterazione abbiamo deciso di aggiungere una lezione introduttiva, per questo chiamata "Lezione 0", per:

- stimolare la creazione di un gruppo classe (gli studenti sono di sezioni diverse e spesso non si conoscono tra loro) facendoli discutere a coppie e gruppi;
- cercare di motivare maggiormente l'importanza della crittografia partendo dal far emergere le idee e le conoscenze pregresse tramite una discussione iniziale aperta;

- introdurre gli studenti a Snap! in un modo semplice e giocoso (creazione di un semplicissimo videogioco), affinché prendessero confidenza con l'ambiente senza dover subito prestare attenzione anche ai concetti di crittografia.

La scansione delle lezioni è presentata di seguito. **In grassetto** sono evidenziate le differenze rispetto alla prima iterazione.

Lezione 0

- **Discussione tra pari: dove e per quali scopi viene usata la crittografia nelle nostre vite; perché è importante nella società**
- Il dibattito sociale sulla crittografia nella messaggistica istantanea
- **Introduzione a Snap!: creazione di un semplice videogioco**
- Cifrario di Cesare (alla lavagna): cifratura, decifratura, **usi oggi (ROT-13)**
- **Compiti a casa: inventa una metodologia di comunicazione sicura**

Lezione 1

- Cifrario di Cesare: cifratura, decifratura, attacco a forza bruta
- Elementi di un crittosistema e principio di Kerckhoffs
- Cifrario di Cesare: attacco con le frequenze
- **Compiti a casa: inventare un crittosistema che resista a forza bruta e frequenze**

Lezione 2

- Cenni ai cifrari polialfabetici ed Enigma
- *One-time pad*: cifratura, decifratura, attacco con frequenze, attacco a forza bruta
- *One-time pad*: Motivazioni alla base della sicurezza perfetta, limiti, problema della distribuzione delle chiavi
- Compiti a casa: quiz su *One-time pad*, riflessione sul problema dello scambio sicuro di una chiave

Lezione 3

- **Esempio di cifrario a trasposizione**
- **Codifica dei caratteri tramite bit, operatore XOR, UNICODE**
- **Cifrario didattico interattivo: semplice rete a sostituzione e permutazione; riflessione su proprietà di confusione e diffusione**

- **AES: cenni su lunghezza della chiave, sicurezza computazionale**
- Compiti a casa: continuare la riflessione sullo scambio sicuro di una chiave tramite un canale insicuro

Lezione 4

- Protocollo di Diffie-Hellman: simulazione con i colori
- Matematica del protocollo di Diffie-Hellman: aritmetica modulare, esponenziale e sua inversione (logaritmo discreto), primi e coprimi (e accenni ai generatori)
- Protocollo di Diffie-Hellman: esempio con numeri, sicurezza computazionale, attacco *person-in-the-middle*
- Crittografia asimmetrica: terminologia, proprietà delle coppie di chiavi pubblica/privata, schemi (non tecnici, ad alto livello) di autenticazione e segretezza, idea intuitiva di funzione unidirezionale (moltiplicazione di primi versus fattorizzazione)
- Unire il tutto: come combinare intuitivamente schemi asimmetrici e simmetrici per l'autenticazione e la segretezza
- Compiti a casa: questionario di soddisfazione, valutazione tramite completamento di riassunto

2.5. Strumenti, attività e metodologie

Strumenti usati

La scuola ospitante usa la suite Google Workspace for Education. Di conseguenza, per le lezioni in videoconferenza della prima iterazione abbiamo usato Google Meet.

In entrambe le iterazioni abbiamo usato Google Classroom per condividere link alle attività, indicazioni e avvisi, materiali e compiti a casa. Abbiamo inoltre usato Google Slides per presentare contenuti e animazioni degli schemi crittografici, ma anche come supporto costante ai vari momenti delle lezioni. Abbiamo usato Google Docs per condividere testi più lunghi e raccogliere riflessioni ed elaborati degli studenti. Inoltre, abbiamo usato Google Forms come strumenti di lavoro per strutturare le attività pratiche con opportune domande-guida, per la consegna dei compiti a casa e, alla fine del corso, per raccogliere le percezioni ed opinioni sul corso degli studenti e le loro risposte nell'attività finale di consolidamento e valutazione.

Snap! è un linguaggio di programmazione visuale a blocchi. Si tratta di una reimplementazione di Scratch (un noto linguaggio a blocchi destinato ai bambini dagli 8 anni in su), ma con molte funzionalità aggiuntive che consentono di utilizzarlo per introdurre in modo semplice ma serio la programmazione informatica a studenti novizi delle scuole secondarie e persino dell'università. Nel nostro caso specifico, abbiamo scelto Snap! perché permette di creare nuovi blocchi oltre a quelli già presenti. A

differenza di quel che accade in Scratch, tali blocchi possono restituire valori (sono cioè funzioni, nel senso informatico del termine) e la loro implementazione (il corpo della funzione) puó essere nascosta agli studenti.

Strumenti creati, attività progettate, metodologie utilizzate

Usando Snap! abbiamo creato una progressione di *playground* crittografici, cioè ambienti digitali in cui gli studenti possono sperimentare alcuni crittosistemi usando la programmazione informatica, circoscritta però a semplici combinazioni di pochi blocchi disponibili. Questi ambienti permettono agli studenti di utilizzare e di provare ad attaccare alcuni crittosistemi significativi (ad esempio, il cifrario di Cesare e il cifrario *One-time pad*) e anche di fare esperienza concreta dei loro limiti (come, ad esempio, la facilità o difficoltà nell'attaccarli, oppure il tempo di elaborazione richiesto). Nello specifico, questi *playground* sono dei progetti Snap! in cui il set di istruzioni (i blocchi) visibili e disponibili è stato limitato.

Sfruttando la possibilità data da Snap! di nascondere alcuni blocchi predefiniti del linguaggio e soprattutto di creare nuovi blocchi personalizzati, per ogni crittosistema abbiamo fornito agli studenti solo i blocchi necessari per cifrare e decifrare messaggi, ed eventualmente, a seconda dell'attività, per condurre possibili attacchi al sistema. Abbiamo individuato specifici meccanismi crittografici su cui volevamo attirare l'attenzione degli studenti e li abbiamo implementati tramite Snap! come blocchi *ad-hoc* del linguaggio. Abbiamo cercato di massimizzare l'espressività di questi nuovi blocchi rispetto allo scenario crittografico, anche riducendo al minimo gli aspetti tecnici legati alla programmazione, in modo da facilitarne la comprensione e l'utilizzo. In virtù di queste scelte, i nostri *playground* possono essere considerati come Teaspoon Languages [43], ossia "linguaggi orientati al compito[1] [... che]: supportano attività di apprendimento che i docenti (tipicamente non di Informatica) vogliono che gli studenti completino; sono linguaggi di programmazione poichè descrivono processi computazionali da far eseguire ad un agente computazionale; si imparano in meno di 10 minuti, in modo da poter essere appresi e usati in un'ora di lezione" [14, nostra traduzione].

Dato che i nostri studenti di liceo non avevano alcuna esperienza pregressa di programmazione, non abbiamo ritenuto fattibile – e nemmeno utile, considerati il contesto del Liceo Matematico e gli obiettivi di alto livello del corso – che effettivamente programmassero (gli algoritmi de)i crittosistemi, ad un livello di astrazione più basso rispetto alla combinazione di blocchi crittografici (il che li avrebbe allontanati dallo scenario crittografico e dal focus sulle idee fondamentali). Tuttavia, volevamo comunque dare agli studenti l'opportunità di costruire la loro conoscenza attraverso la manipolazione concreta di oggetti computazionali legati alla crittografia. In particolare, invece di mettere gli studenti in micro-mondi *à la Papert* [30], che di solito si basano su linguaggi di programmazione *general-purpose* come LOGO – che, anche quando pensati per scopi didattici, richiedono un tempo significativo di apprendimento – abbiamo sviluppato e realizzato mini-linguaggi molto più semplici (e quindi più fa-

[1] *task-specific* nel testo originale.

cili da usare), circoscritti (quindi non *general-purpose*) ai nostri specifici obiettivi di apprendimento [15]. Per esempio, nel *playground* in cui gli studenti possono provare ad attaccare il cifrario di Cesare usando le frequenze, i blocchi disponibili sono: calcolare le frequenze delle lettere in un testo, ordinare una tabella (ossia, una matrice di lettere e le relative frequenze) per frequenza, rappresentare i dati di una tabella con un istogramma, e la tabella delle frequenze medie delle lettere in latino (fig. 1).

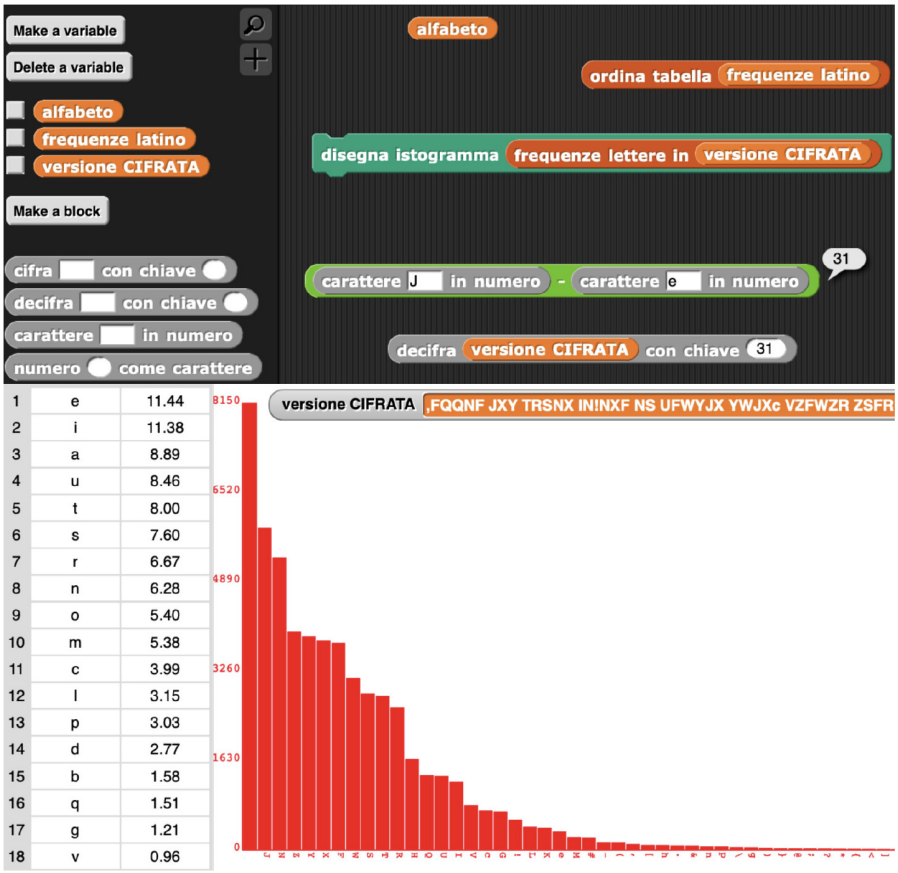

Figura 1: *Playground* Snap! per l'attacco con le frequenze al cifrario di Cesare

Tutti i *playground* che abbiamo creato sono disponibili per essere liberamente esplorati ed utilizzati [24]. Gli studenti, seguendo le indicazioni operative che abbiamo di volta in volta condiviso con loro, si sono cimentati in piccole sfide in cui dovevano cifrare, decifrare o attaccare un certo sistema combinando i pochi e specifici blocchi Snap! forniti nel *playground*. L'obiettivo più generale di queste sfide era quello di far sperimentare agli studenti i crittosistemi, per favorire una comprensione radicata

nell'esperienza del loro funzionamento e dei loro limiti. In più, abbiamo cercato di contestualizzare queste attività pratiche in scenari significativi per gli studenti. Per esempio, le attività sul cifrario di Cesare riguardavano la possibilità di decifrare e quindi scoprire un testo latino[2] (la versione che sarebbe stata usata per la verifica) che il loro docente, in quarantena per il COVID, doveva comunicare segretamente al suo supplente senza che gli studenti potessero intercettarlo.

Poichè gli studenti dei corsi erano del tutto nuovi alla programmazione informatica, abbiamo ritenuto che fosse una sfida troppo impegnativa per loro (e non opportuna in un corso breve e con altri obiettivi) programmare un protocollo crittografico come lo scambio di chiave di Diffie-Hellman. Di conseguenza abbiamo deciso di adottare un approccio *unplugged* al fine di sviluppare un'attività comunque concreta, che portasse all'effettiva generazione di una chiave segreta condivisa ma che mantenesse un forte legame con gli aspetti formali di questo rivoluzionario protocollo. L'approccio *unplugged* puó infatti aiutare a comprendere ad alto livello il funzionamento di algoritmi informatici significativi, facendo in modo che gli studenti possano eseguirli "in prima persona" attraverso attività cinestetiche e divertenti [2].

Nella letteratura e nei materiali didattici analizzati, abbiamo trovato almeno due modi tradizionalmente usati per spiegare il funzionamento del protocollo di Diffie-Hellman per la generazione di una chiave segreta condivisa (e.g., [42]): il mixaggio dei colori, più evocativo ma semplificato, e l'esecuzione passo passo dell'algoritmo con numeri piccoli, più preciso e fedele ma anche più complesso da comprendere in prima battuta. Abbiamo cercato di combinare queste due modalità, in particolare sviluppando una serie di fasi didattiche per passare gradualmente dalla metafora dei colori al funzionamento matematico dell'algoritmo.

Il cuore della nostra strategia è un'attività *unplugged* a coppie, basata su un progetto Snap! solo eseguibile (ossia dal codice non modificabile nè esplorabile, che quindi funziona come app interattiva) che guida gli studenti nell'esecuzione passo passo del protocollo di Diffie-Hellman rappresentato con il mixaggio dei colori [24]. Il mixaggio dei colori non è stato realizzato con i classici algoritmi additivi o sottrattivi, ma si basa sui calcoli effettivi (pur se inizialmente nascosti agli studenti) del protocollo di Diffie-Hellman su numeri piccoli (da 0 a 99 – numeri che rappresentano i colori disponibili in Snap!). Usando il progetto Snap! come una app interattiva, gli studenti, a coppie, sono stati in grado di generare un colore segreto condiviso. Le comunicazioni all'interno di ogni coppia – necessarie per concordare un colore iniziale e poi per lo scambio dei rispettivi colori calcolati intermedi – hanno avuto luogo nella chat pubblica della videoconferenza con cui venivano trasmesse le lezioni. La chat è stata usata per rappresentare, e lo ha fatto in modo efficace, un canale insicuro: tutti i partecipanti alla lezione potevano "ascoltare" su quel canale (cioè leggere la chat) senza restrizioni (vedi fig. 2).

Dopo che gli studenti hanno sperimentato in prima persona il funzionamento di alto livello del protocollo di Diffie-Hellman, abbiamo illustrato gli strumenti matematici essenziali per il suo funzionamento (vedi 2.4). Poi, abbiamo mostrato l'esecuzione passo passo del protocollo tra due parti comunicanti, servendoci di un'animazione [24].

[2] Tutti gli studenti dei nostri corsi avevano Latino tra le loro materie curricolari.

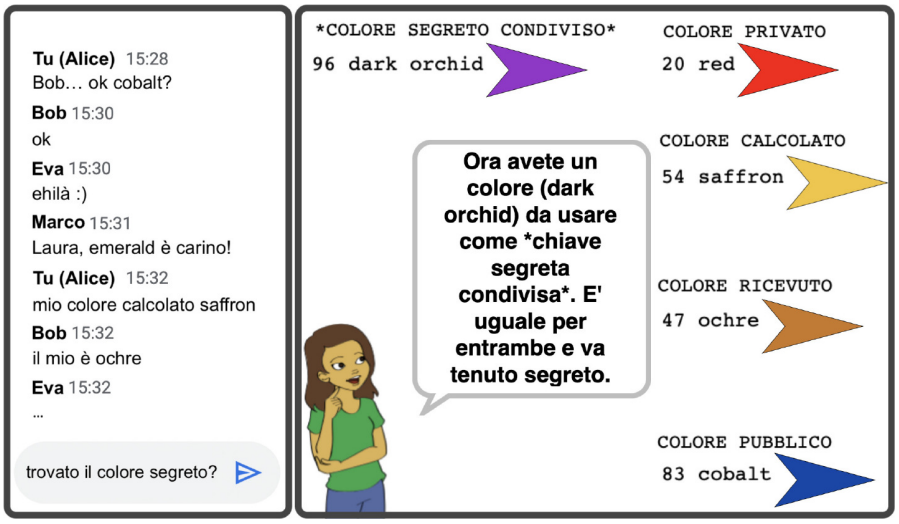

Figura 2: Scenario dell'attività su Diffie-Hellman: chat delle lezioni e app di supporto

L'animazione mostra la corrispondenza tra i colori e i numeri (piccoli, da 0 a 99) che li rappresentano in Snap!, rivelando i calcoli effettivi che regolano il mixing dei colori, appena sperimentato dagli studenti nell'attività *unplugged*.

Data la breve durata del corso, per gli schemi più avanzati basati sulla crittografia a chiave pubblica ci siamo limitati ad una metodologia didattica sempre interattiva e dialogata, ma più frontale. Abbiamo ritenuto che le attività pratiche – efficaci nel supportare l'apprendimento ma che richiedono necessariamente tempi più lunghi – fossero strategiche in particolare nella prima parte del corso, per facilitare a tutti la comprensione dei concetti e dei meccanismi di base, ma anche per condividere da subito, *in-context*, un linguaggio specifico e una serie di ragionamenti che pongono le basi per la comprensione dei più complessi crittosistemi moderni. Su queste fondamenta più solide, ci siamo dovuti limitare a illustrare ad alto livello e discutere i meccanismi della crittografia asimmetrica, i loro scenari di utilizzo e le ragioni della loro importanza. D'altra parte, visti gli obiettivi generali del corso (ossia la comprensione delle idee fondamentali della crittografia e soprattutto la consapevolezza di come l'interazione dei relativi schemi crittografici permetta oggi sicurezza e autenticazione sicure, veloci e alla portata di tutti), abbiamo deciso di aumentare il passo e il livello di astrazione per riuscire a completare la panoramica. Infatti riteniamo che un quadro completo e "funzionante", seppur semplificato e astratto, sia necessario per coinvolgere gli studenti su un piano di realtà che si leghi alle loro esperienze quotidiane con la crittografia (ad esempio, le comunicazioni riservate con gli amici, la partecipazione nei social media e l'utilizzo di servizi digitali a pagamento).

Abbiamo illustrato agli studenti che è possibile creare due chiavi, una pubblica

e una privata, legate dalla proprietà che ció che è stato chiuso (cifrato) con una chiave puó essere aperto (decifrato) solo con l'altra. Ci siamo limitati a suggerire l'intuizione matematica su cui si basa questa crittografia asimmetrica: la chiave pubblica è legata alla moltiplicazione di due numeri primi molto grandi (facile), mentre la chiave privata è legata alla fattorizzazione del loro (molto grande) prodotto (difficile). Abbiamo chiesto agli studenti di immaginare come utilizzare la coppia di chiavi pubblica e privata per realizzare una comunicazione segreta. Poi, li abbiamo guidati verso il concetto di autenticazione, concetto che emerge per la prima volta qui, e abbiamo illustrato la sua realizzazione con lo schema asimmetrico. Abbiamo quindi discusso ad alto livello l'utilizzo speculare delle due chiavi per ottenere segretezza o autenticazione, e anche come combinarne l'uso per ottenere entrambe le proprietà. Per supportare la comprensione degli schemi della crittografia asimmetrica, abbiamo sviluppato animazioni [24] di semplici scenari di comunicazione. In queste animazioni abbiamo usato due personaggi della nota serie televisiva *Power Rangers*, in modo che le azioni e i messaggi delle diverse parti fossero evidenti attraverso i loro rispettivi colori. Le varie scene di queste animazioni hanno scandito e supportato l'introduzione dei concetti crittografici, favorendo le riflessioni grazie ai loro elementi concreti. Terminata l'analisi passo passo (ma ad alto livello) dei due schemi di comunicazione proposti, specifiche animazioni conclusive hanno permesso agli studenti di riepilogare visivamente il funzionamento complessivo degli schemi asimmetrici per l'autenticazione e la segretezza.

I moderni cifrari a blocchi a chiave simmetrica (come AES) erano stati trattati nella prima iterazione solo nei compiti a casa, tramite una lettura sull'origine di DES e con un semplice esercizio di manipolazione di bit. Poichè questo approccio era risultato ostico e non particolarmente coinvolgente, abbiamo deciso di utilizzare parte del tempo in più per far comprendere meglio agli studenti i meccanismi alla base delle moderne reti a sostituzione e permutazione. Abbiamo quindi realizzato, tramite un foglio di calcolo Google, una versione semplificata di una rete di quel tipo, che lavora su 4 blocchi da 4 bit ciascuno, con tre *round* di sostituzione e due di permutazione. Nonostante le dimensioni ridotte della rete, il foglio di calcolo (in cui gli studenti possono modificare la chiave e il messaggio [24] e osservare in diretta i cambiamenti sui bit che attraversano la rete) permette di vedere in modo semplice e interattivo l'"effetto valanga" suscitato sul messaggio cifrato da minimi cambiamenti nei bit della chiave o del messaggio in chiaro.

L'altra modifica rilevante rispetto alla prima iterazione riguarda più in generale gli altri compiti a casa. Nella seconda iterazione infatti i primi compiti a casa hanno stimolato gli studenti a inventare i propri crittosistemi (il primo completamente libero, poi uno che resistesse ad attacchi a frequenze e forza bruta, infine un modo per scambiarsi una chiave segreta in modo sicuro). Questi compiti hanno sempre preceduto la spiegazione delle possibili soluzioni trovate, nel corso della storia, dalla crittografia. In questo modo, gli studenti hanno sperimentato ancora di più i limiti e le difficoltà cruciali affrontate nel design dei crittosistemi (e quindi sentito la necessità di nuovi schemi crittografici che li superassero). Le lezioni che seguivano questi compiti a casa (in cui spesso gli studenti hanno trovato soluzioni creative, anche se – come era ragionevole – raramente sicure come richiesto) sono iniziate sempre con discussioni molto parteci-

pate, perfette come trampolino di lancio per introdurre uno schema "emblematico" che in qualche modo risolvesse le problematiche rilevate nei tentativi degli studenti (e.g., *One-time pad*, Diffie-Hellman).

3. Confronto della proposta con la letteratura sulla didattica della crittografia

3.1. Quadri di riferimento internazionali

Nel 2017 le più importanti associazioni americane e internazionali di informatici e ingegneri informatici (tra cui ACM – *Association for Computing Machinery*, e IEEE *Computer Society*) hanno sviluppato il *Cybersecurity Curricula 2017* [17], che raccoglie le linee guida per i corsi di laurea in *cybersecurity*. Il documento, chiaramente rivolto a insegnamenti specialistici, include la crittografia nella prima unità di apprendimento, ritenuta necessaria per porre le basi per gli apprendimenti successivi [17, p. 24]. Tra i contenuti di crittografia indicati nel documento, c'è una vastissima sovrapposizione con quelli insegnati nel nostro corso:

- **concetti di base** come, ad esempio, funzioni di cifratura e decifratura, autenticazione, chiave simmetrica versus asimmetrica, sicurezza perfetta (e.g., *One-time pad*), sicurezza computazionale, etc.;
- **background matematico** come, ad esempio, aritmetica modulare, radici primitive e logaritmo discreto, test di primalità versus fattorizzazione di interi grandi, etc.;
- **cifrari storici** come, ad esempio, cifrari a trasposizione, a sostituzione (e.g., Vigenère, ROT-13), macchina Enigma, etc.
- **cifrari simmetrici** come, ad esempio, cifrari a blocchi quali DES, AES, etc.;
- **cifrari asimmetrici** e concetti quali complessità computazionale, funzioni *one-way*, protocollo di Diffie-Hellman, idea di RSA, etc.

Per quanto riguarda più specificamente la formazione pre-universitaria (K-12: dalla scuola dell'infanzia, K, fino al grado 12, ossia all'incirca la quarta superiore – comprendente quindi il periodo della scuola dell'obbligo italiana), gli standard [8] suggeriti dall'associazione dei docenti di informatica americani (CSTA), insieme ad alcune delle già citate associazioni di categoria, includono la *cybersecurity* come importante argomento a tutti i livelli scolastici. In particolare, nel 'Livello 2', quello relativo agli studenti tra gli 11 e i 14 anni, lo standard indica che gli studenti dovrebbero

> applicare diversi metodi di crittografia per modellare la trasmissione sicura delle informazioni. I metodi crittografici possono essere semplici come la sostituzione di lettere o più complicati, come i moderni metodi usati per proteggere le reti e Internet. Gli studenti dovrebbero codificare e decodificare messaggi usando vari metodi di crittografia e dovrebbero comprendere i diversi livelli di complessità utilizzati per nascondere o proteggere

> le informazioni. Per esempio, gli studenti potrebbero cifrare i messaggi usando metodi come il cifrario di Cesare o la steganografia (cioè nascondere messaggi all'interno di un'immagine o di altri dati). Possono anche studiare modelli di metodi più complicati, come la crittografia a chiave pubblica, attraverso attività *unplugged* [8, 2-NI-06, nostra traduzione].

Il percorso che abbiamo progettato è in linea con questi standard, sia per quanto riguarda i contenuti di base, sia per le metodologie adottate (e.g., attività *unplugged* per i complessi cifrari moderni).

3.2. Didattica della crittografia nelle conferenze di didattica dell'Informatica

Una rassegna [41] delle pubblicazioni in didattica della *cybersecurity* nelle due più importanti conferenze ACM tra il 2010 e il 2019 ha rilevato che la ricerca si focalizza prevalentemente sulla formazione universitaria negli Stati Uniti. Solo 14 dei 71 lavori considerati includono argomenti di crittografia. La maggior parte di essi riguarda la prospettiva più ampia della *cybersecurity*, e dunque la crittografia è solo uno dei tanti argomenti considerati (ad esempio, [4, 10, 36, 39]); la crittografia è spesso vista come argomento tecnico e strumentale più che essere trattata nell'ottica dal punto di vista dei suoi principi fondanti. Pochissimi lavori (ad esempio, [5, 16]) riguardano specificamente la crittografia.

Vista l'importanza e l'innovatività del tema nella formazione pre-universitaria, abbiamo deciso di condurre un'estensiva rassegna dei lavori che trattano l'insegnamento della crittografia nel contesto scolastico. Abbiamo considerato sia lavori incentrati solo sulla crittografia, sia lavori che riguardano più estesamente la *cybersecurity*, solo quando però la crittografia era inclusa in maniera rilevante.

Nelle sezioni seguenti confrontiamo le principali tendenze emerse da questa rassegna con la nostra proposta.

3.3. Attività pratiche

Le proposte didattiche che abbiamo trovato includono spesso attività pratiche, molte volte situate in contesti motivanti e realistici. Per esempio: comprendere uno scambio email sicuro tramite simulazioni con diversi cifrari, classici e moderni [12]; realizzare, tramite linguaggi di programmazione a blocchi, comunicazioni sicure tra robot nel contesto di campi estivi sull'insegnamento della *cybersecurity* e del pensiero computazionale [21,44]; costruire un social-network "giocattolo" attraverso cui apprendere concetti di sicurezza, crittografia e programmazione [45]; simulare ad alto livello sfide di sicurezza e crittografia attraverso l'uso di specifici strumenti web [33].

In generale, nel contesto della crittografia e della *cybersecurity*, si ritiene che le attività attive e partecipative (*hands-on*) e *inquiry-based* possano migliorare l'autoefficacia degli studenti e le loro capacità di *problem solving* [20].

Il nostro percorso è guidato da domande attuali e rilevanti per gli studenti, ad esempio "Le chat di WhatsApp sono davvero sicure?" e "Di quali conoscenze abbi-

amo bisogno per convincerci che lo siano, e in generale per capire come funziona la sicurezza nella messaggistica istantanea?" Inoltre, tutte le nostre attività sono contestualizzate in situazioni significative per gli studenti (per esempio "Riesci a decifrare il testo della verifica che i tuoi insegnanti si sono scambiati in modo cifrato?"). Questo tipo di attività fa leva sulla dimensione di sfida e avventura che la crittografia naturalmente offre [12, 23].

Visti i contenuti suggeriti nella letteratura e nei curricoli, e considerando il nostro target e la prospettiva culturale adottata (i.e., la formazione scolastica non mira a formare professionisti ma a dare strumenti per capire il mondo, vedi sezione 1), ci poniamo l'obiettivo di rendere comprensibili le idee e principi fondamentali della crittografia. Pertanto, il nostro percorso si concentra su alcuni crittosistemi strategici, selezionati per far comprendere i principi fondamentali, approfondendo i dettagli tecnici solo quando sono strumentali alla comprensione di tali principi. Questa prospettiva ha guidato la selezione dei contenuti, lo sviluppo del percorso, la progettazione delle attività e lo sviluppo e l'uso dei relativi strumenti.

3.4. Strumenti di visualizzazione e programmazione a blocchi

Nel tempo sono stati sviluppati diversi strumenti di visualizzazione e di simulazione interattiva per insegnare algoritmi e sistemi crittografici (quali Cesare, Vigenère, DES, AES, RSA, SHA [1,27,34,35, per esempio]). Solitamente queste simulazioni mostrano come funzionano i cifrari, le loro debolezze e quindi i possibili attacchi. Spesso queste simulazioni sono accurate ma troppo ricche di dettagli tecnici per studenti di scuola superiore, per di più novizi di crittografia. Inoltre, l'interattività è spesso limitata all'inserimento del messaggio da cifrare, al procedere passo-passo e al più a visualizzare brevi spiegazioni; in generale è limitata alla modifica di alcuni parametri della simulazione.

In contrasto a queste proposte, nel nostro percorso gli studenti *programmano parte della propria esperienza di apprendimento*. Come detto, gli studenti manipolano liberamente e combinano oggetti digitali (i.e., blocchi di un linguaggio di programmazione visuale) che rappresentano importanti concetti di crittografia. Ad esempio, gli studenti combinano i blocchi per calcolare e visualizzare le frequenze delle lettere in un testo cifrato, in modo da poter eseguire un attacco basato sulle frequenze (vedi fig. 1).

Il nostro approccio condivide – e cala nel contesto della formazione scolastica – la visione di van der Linden e colleghi [40], che suggeriscono di usare la metafora dei blocchi (tratta dai linguaggi di programmazione visuale) per rappresentare funzionalità crittografiche e permettere agli sviluppatori di combinarle in maniera efficace e sicura.

Un approccio simile al nostro è stato proposto anche da McAndrew [28], che utilizza i CAS (*Computer Algebra Systems*) per far implementare agli studenti algoritmi crittografici classici e moderni, in modo che possano comprenderne meglio il funzionamento e i possibili attacchi programmandoli ad alto livello.

3.5. Attività *unplugged*

Alcuni autori hanno suggerito e implementato attività di tipo *unplugged* per insegnare la crittografia. In queste attività gli studenti sperimentano ad alto livello algoritmi di cifratura e decifratura, protocolli e attacchi.

Per esempio, Bell (uno dei primi ad aver realizzato attività *unplugged*) propone di simulare una funzione unidirezionale (concetto fondamentale nella crittografia a chiave pubblica) con la ricerca di un *perfect dominating set* su un grafo, sui cui nodi gli studenti effettuano elementari calcoli aritmetici per scambiarsi un numero segreto [3]. Konak [19] ha proposto semplici attività *unplugged* per spiegare crittosistemi classici e moderni con carta, penna, forbici, scatole e lucchetti. Fees e colleghi [11] hanno reso concreta la metafora dei colori usata per lo scambio di chiave di Diffie-Hellman facendo mescolare agli studenti coloranti alimentari, in modo da generare una chiave segreta condivisa.

Nella prima iterazione del nostro corso – tenutasi esclusivamente a distanza – abbiamo trovato il modo di realizzare una attività *unplugged* sul protocollo di Diffie-Hellman sfruttando lo strumento di comunicazione a distanza (in particolare, la chat pubblica delle lezioni online), mantenendo però lo spirito delle attività *unplugged* (come discusso nella sezione 2.5). Un approccio simile puó essere ritrovato nel lavoro di Greenlaw e colleghi [13], che hanno usato la bacheca di una classe virtuale per dimostrare come funziona un attacco *person-in-the-middle* a un crittosistema a chiave pubblica.

4. Raccolta e analisi dati

Alla fine di ciascuna iterazione del corso, abbiamo chiesto agli studenti di compilare due moduli Google per raccogliere i loro feedback e valutare il loro apprendimento. Non erano previsti voti per queste attività finali.

Nella prima iterazione, i questionari sono stati compilati dagli stessi 14 studenti (su 15). Nella seconda iterazione, dagli stessi 11 studenti (su 13).

4.1. Valutazione dell'apprendimento

Abbiamo preparato un riassunto di circa 2000 parole [24] delle idee e dei concetti importanti affrontati nel corso, identificando nel testo diversi passaggi chiave. Per ognuno di essi, gli studenti dovevano scegliere tra il riempimento giusto e un'alternativa sbagliata. Volevamo che l'attività fosse anche un'opportunità di ripasso per gli studenti, e quindi l'abbiamo strutturata come un "racconto" che riassume i contenuti più importanti del corso.

Il testo originale (usato alla fine della prima iterazione) includeva 43 passaggi chiave in cui effettuare una scelta tra due opzioni. Nella seconda iterazione, il testo è rimasto sostanzialmente inalterato, al netto di minime modifiche stilistiche e dell'aggiunta di poche frasi relative ai contenuti aggiuntivi. Dunque, nella sua seconda versione (quella disponibile nei materiali del corso [24]) i passaggi chiave sono

diventati 46.

Prima iterazione
Dal punto di vista degli apprendimenti, i risultati sono stati positivi: su 43 scelte da effettuare, la media di quelle corrette è stata 32,5, la mediana 34, con un intervallo tra 17 e 41 risposte corrette.

Seconda iterazione
I risultati della seconda iterazione sono stati ancora più positivi: su 46 scelte, la media di quelle corrette è stata 40,7, la mediana 40, con un intervallo decisamente più piccolo, tra 35 e 45 risposte corrette.

4.2. Gradimento

In entrambe le iterazioni, la partecipazione è stata alta.

Prima iterazione
Su 14 studenti che hanno risposto al questionario, 13 hanno seguito la lezione uno, 12 la lezione due, 11 la lezione tre e 12 la lezione quattro.

Seconda iterazione
Su 11 studenti che hanno risposto al questionario, tutti hanno seguito la lezione zero, 10 la lezione uno, 7 la lezione due, 10 la lezione tre e 9 la lezione quattro.

Feedback sull'esperienza

Nel questionario di gradimento, abbiamo chiesto agli studenti feedback sulla loro esperienza di apprendimento e sull'impatto che il corso ha avuto sulla loro percezione di informatica, matematica e crittografia. Era obbligatorio rispondere a tutte le domande.

La durata del corso è risultata adeguata per la stragrande maggioranza degli studenti di entrambe le iterazioni (vedi fig. 3).

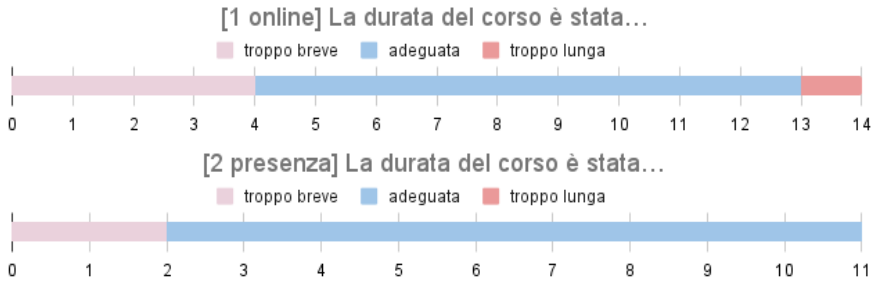

Figura 3: Valutazione della durata del corso

Per quanto riguarda le attività, la maggior parte degli studenti le ha trovate chiare, interessanti, utili per la propria crescita personale e per capire il mondo. Al-

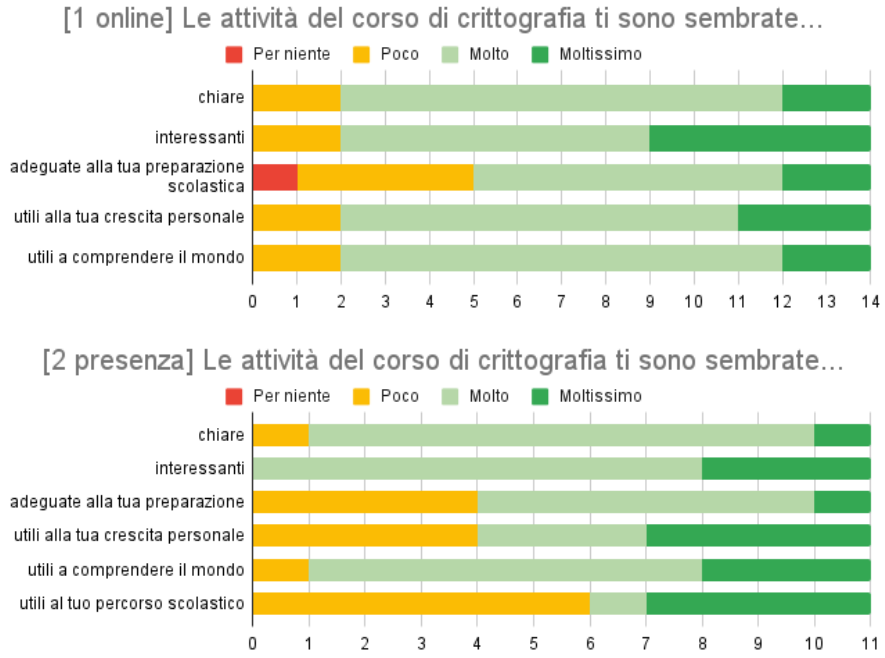

Figura 4: Valutazione generale delle attività

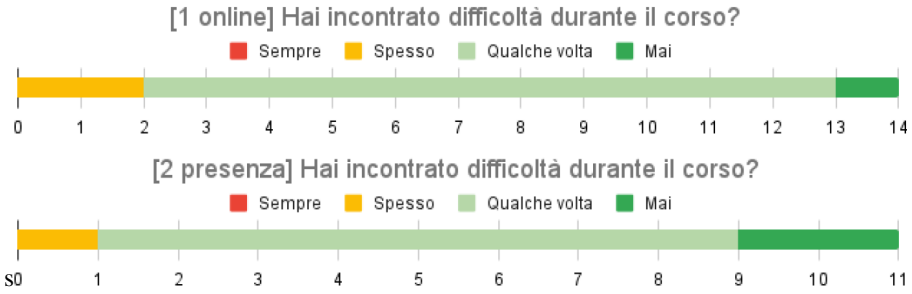

Figura 5: Autovalutazione delle difficoltà incontrate

cuni studenti hanno segnalato che la loro preparazione scolastica pregressa non era del tutto adeguata alle attività del corso (vedi fig. 4). Da notare che nella seconda iterazione è stata chiesta l'utilità per il proprio percorso scolastico, l'unica dimensione su cui la maggioranza degli studenti non si è espressa positivamente.

Relativamente alle difficoltà del corso, in entrambe le iterazioni pochissimi stu-

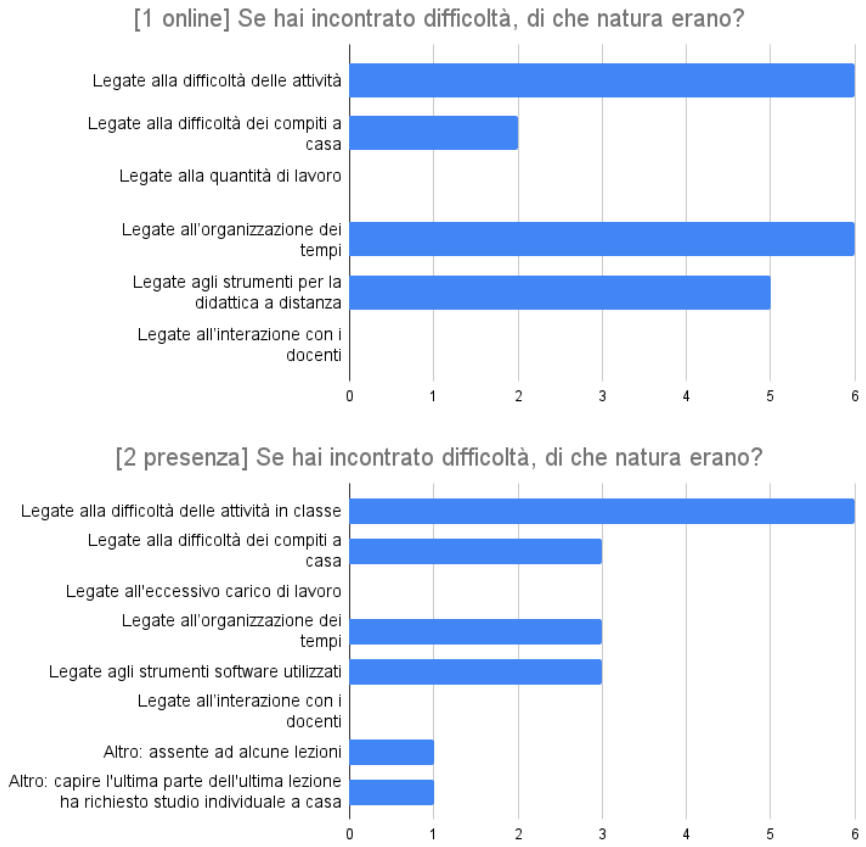

Figura 6: Natura delle difficoltà incontrate

denti hanno dichiarato di aver incontrato ‘Spesso’ difficoltà durante il corso, mentre la quasi totalità solo ‘Qualche volta’ (vedi fig. 5).

In entrambe le iterazioni, la maggior parte delle difficoltà segnalate (se ne poteva scegliere più di una) è relativa alla complessità delle attività svolte in classe. Per quanto riguarda invece tempi e organizzazione, la seconda iterazione ha registrato un netto miglioramento, probabilmente dovuto alla lezione aggiuntiva e al contesto in presenza (vedi fig. 6).

Relativamente all’interazione con i docenti e al supporto ricevuto, in entrambe le iterazioni la soddisfazione è stata estremamente alta (vedi fig. 7).

Anche la soddisfazione complessiva sul corso è stata molto alta in entrambe le iterazioni; solo uno studente della prima iterazione ha detto di essere ‘Poco’ soddisfatto

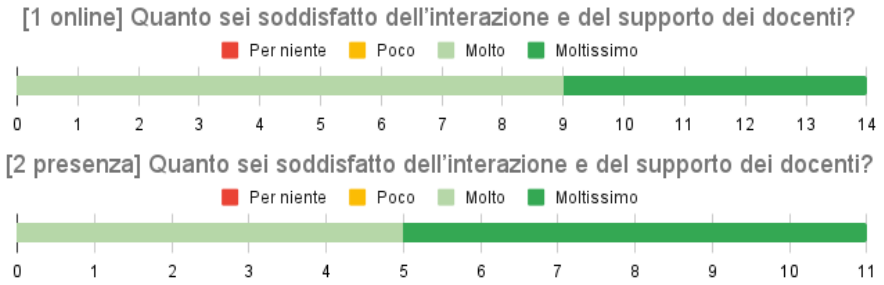

Figura 7: Valutazione del rapporto coi docenti

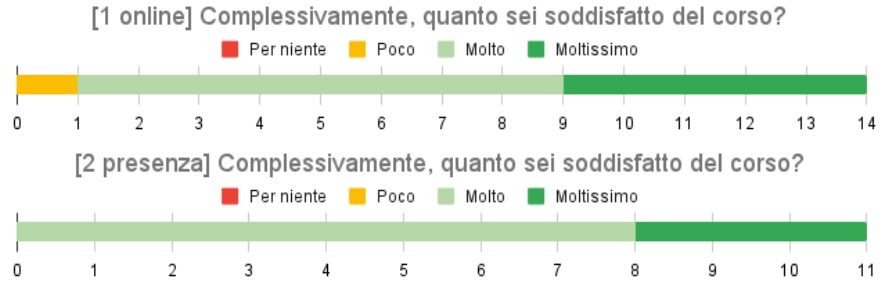

Figura 8: Soddisfazione generale sul corso

(vedi fig. 8). A testimonianza del gradimento elevato, 13 su 14 studenti della prima iterazione, e tutti gli 11 della seconda, consiglierebbero il corso ad un amico.

Andando più nello specifico, abbiamo chiesto agli studenti la loro opinione su specifici strumenti e metodologie per meglio valutarne efficacia e gradimento. Rimandiamo alle figure (da 9 a 12) per una panoramica dettagliata, ma riportiamo qui le considerazioni che ne emergono che riteniamo più rilevanti.

Le attività di crittografia a blocchi con i *playground* Snap! sono state ritenute 'Molto' o 'Moltissimo' utili e coinvolgenti, anche se poco meno della metà degli studenti non le ha trovate facili (vedi fig. 9).

Nella seconda interazione, l'essere in presenza ci ha permesso di condurre discussioni molto partecipate e che ci sono sembrate stimolanti, impressione confermata dalla quasi totalità degli studenti che le hanno ritenute utili e coinvolgenti (vedi fig. 10).

Visto che nella prima iterazione i compiti a casa non sono stati ritenuti particolarmente coinvolgenti, i compiti a casa della seconda iterazione sono stati completamente ripensati. L'opinione degli studenti mostra un chiaro cambio di percezione a favore delle nuove proposte (vedi fig. 11).

L'attività *unplugged* di Diffie-Hellman è stata apprezzata 'Moltissimo' dagli

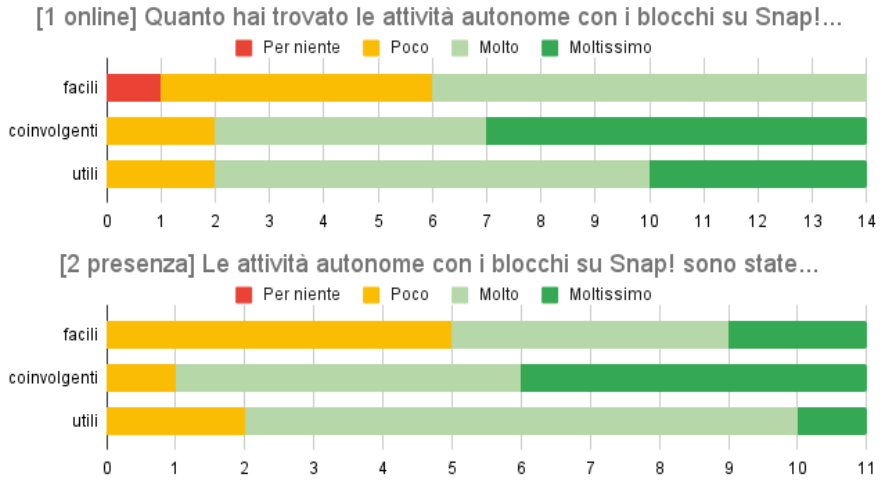

Figura 9: Valutazione delle attività con i *playground* Snap!

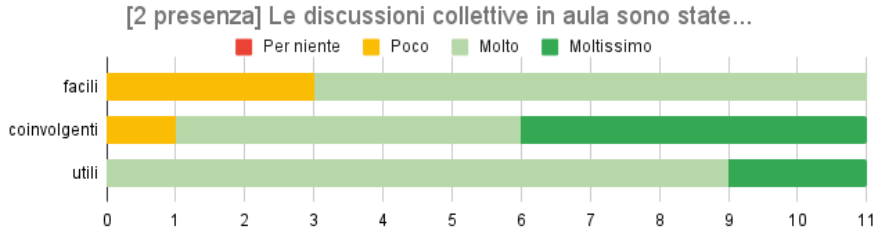

Figura 10: Valutazione delle discussioni in aula

studenti della prima iterazione (vedi fig. 12). Un malfunzionamento della rete della scuola non ha consentito di replicarla nella seconda iterazione, perció siamo stati costretti a mostrarla in azione coinvolgendo solo due studenti. Di conseguenza, la relativa domanda non è stata inclusa nel questionario.

Utilità percepita

Per quanto riguarda l'utilità percepita del corso, la maggior parte degli studenti di entrambe le iterazioni lo ha trovato utile per capire meglio la crittografia, di cosa si occupa e il suo ruolo nella società. È anche migliorata la comprensione del ruolo di informatica e matematica nella società. Questi aspetti sono risultati particolarmente positivi nella seconda iterazione. Più in generale l'utilità percepita è stata più alta nella seconda iterazione.

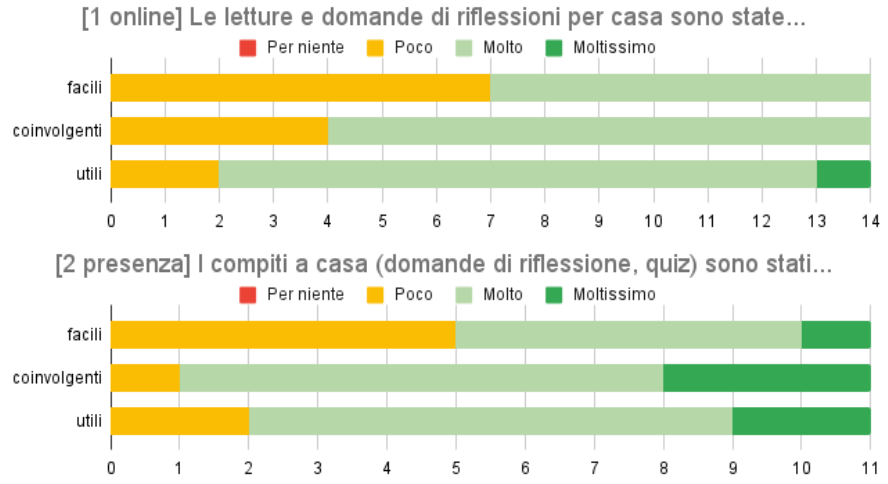

Figura 11: Valutazione dei compiti a casa

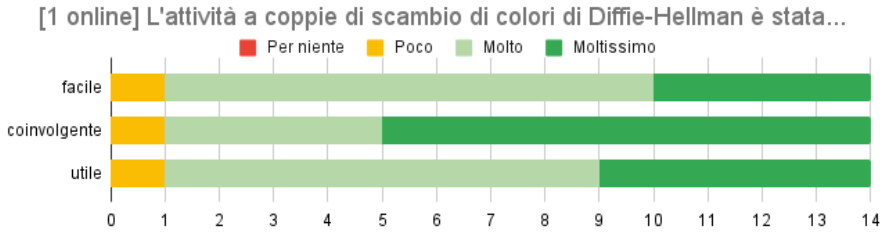

Figura 12: Valutazione dell'attività a coppie su Diffie-Hellman

In entrambe le iterazioni, infine, se è vero che il corso ha stimolato l'interesse per informatica e crittografia in circa $2/3$ degli studenti, solo meno della metà ha percepito un aumento di interesse per la matematica (vedi fig. 13).

Commenti liberi

Alla fine del questionario di gradimento, abbiamo dato la possibilità agli studenti di esprimere liberamente osservazioni, commenti, proposte ("Uno spazio libero per scriverci quello che ti va :-)").

Prima iterazione
Tutti i commenti, tranne uno, sono stati positivi. La maggior parte degli studenti ha affermato che le attività sono state interessanti e divertenti (ad esempio, "Mi è piaciuto

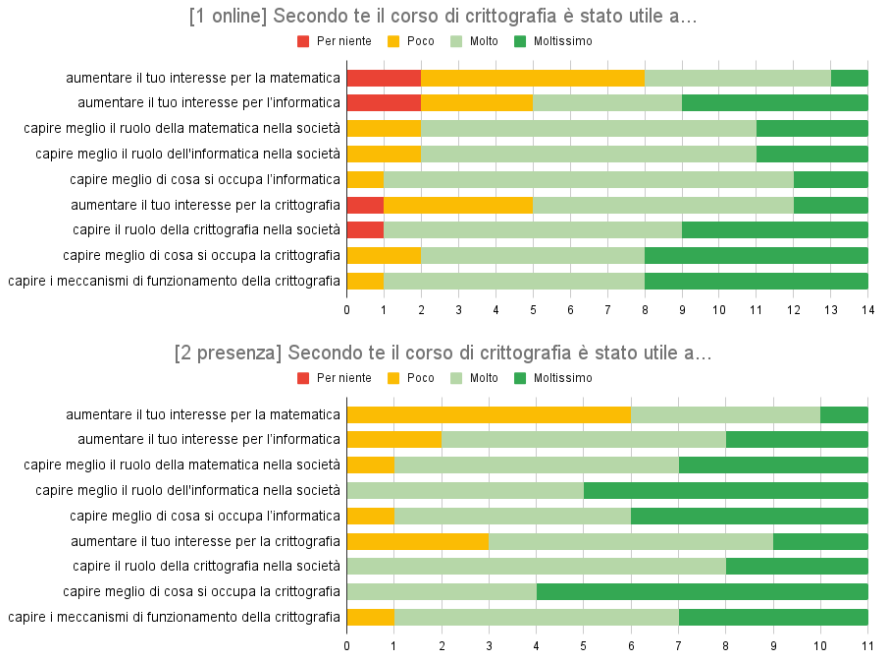

Figura 13: Autovalutazione degli effetti del corso

il fatto che tramite Snap! abbiamo potuto giocare e sperimentare con la crittografia"). Secondo loro, le lezioni sono state coinvolgenti e ben organizzate, anche a distanza. Uno studente ha scritto: "mi sono divertito e ho apprezzato molto che ci sia stato spazio per il dibattito. Ho imparato molto e mi sarebbe piaciuto farlo per più tempo". Anche altri studenti avrebbero voluto che il corso durasse di più. Uno studente avrebbe voluto approfondire la programmazione con Snap! oltre i confini delle attività con i *playground*.

L'unico commento non positivo riguarda difficoltà di comprensione; lo studente avrebbe voluto più "schemi e animazioni prima di passare ad un lavoro più pratico".

Ci fa piacere riportare che uno studente ha usato un *playground* sul cifrario di Cesare per cifrare (con una chiave a noi sconosciuta) il suo commento positivo.

Seconda iterazione

Anche i commenti aperti della seconda iterazione sono stati tutti molto positivi e hanno confermato l'apprezzamento per il corso.

Il corso è stato considerato stimolante (tanto da indurre qualche studente a proseguire autonomamente lo studio della crittografia), "più interessante di quanto potessi pensare", "perché ti insegna cose che non ti insegnano a scuola".

5. Risultati e osservazioni

5.1. Metodologie usate e risultati nelle due iterazioni

Le attività sul protocollo di Diffie-Hellman sono state molto apprezzate: la simulazione del canale insicuro attraverso la chat pubblica è stata percepita, come da nostra intenzione, come una metafora utile a comprendere il protocollo; gli studenti hanno trovato l'attività coinvolgente e divertente (fig. 12), confermando le nostre impressioni positive. La consideriamo un'attività "*remote-unplugged*" perché ha molte delle caratteristiche di un'attività *unplugged* (ossia: presenta concetti e algoritmi fondamentali dell'informatica; si basa sull'"imparare facendo"; divertente; cooperativa; autocontenuta; resiliente agli errori degli studenti – si veda [7]) eccetto per il fatto che è stata erogata tramite elaboratori elettronici. Nel nostro caso, però, gli elaboratori sono stati soltanto un mezzo di comunicazione piuttosto che uno strumento informatico necessario all'attività stessa. Il progetto interattivo Snap! ha reso questa attività sul protocollo di Diffie-Hellman concreta e facile da seguire, rivelandosi un'ottima soluzione per il contesto remoto della prima iterazione del corso. Come detto, non abbiamo potuto replicare questa attività in presenza in occasione della seconda iterazione per un malfunzionamento della rete della scuola ospitante. In una prospettiva di sola presenza, si potrebbe riadattare affinché diventi *unplugged* a tutti gli effetti, a patto di trovare un adeguato mezzo di comunicazione che rappresenti il canale insicuro (e.g., una lavagna su cui tutti possono leggere e scrivere).

Tutto il corso è stato impostato attorno a discussioni collettive, guidate dai dubbi e dalle intuizioni degli studenti. Lo spazio dato a queste interazioni è stato molto apprezzato (vedi 4.2). Confrontando le due iterazioni, quella in presenza ha visto una partecipazione più ampia alle discussioni, imputabile a nostro giudizio a diversi fattori:

- la "Lezione 0", iniziata con discussioni a coppie e a gruppi prima ancora di introdurre qualunque concetto di crittografia, ha subito impostato una modalità di lavoro basata sull'interazione;
- essere in presenza ha favorito un'interazione più naturale con i docenti e tra gli studenti (che all'inizio del corso non si conoscevano, provenendo da sezioni diverse);
- compiti a casa più creativi e aperti hanno favorito dibattiti sulle diverse proposte degli studenti.

I *playground* Snap! hanno funzionato bene – per i crittosistemi più semplici come il cifrario di Cesare – per fare esperienza e capire gli elementi di un sistema crittografico, alcuni dei possibili attacchi, e le sue principali limitazioni (ad esempio, il tempo di calcolo richiesto). Tuttavia, specialmente nella prima iterazione, alcuni studenti hanno trovato difficili i *playground* più avanzati. In questo caso ha probabilmente pesato il contesto remoto. Non poter "girare tra i banchi" ha portato ad una significativa "cecità del docente", rendendoci difficile agire da facilitatori e fornire agli studenti una *guida ottimale* [37] durante l'esplorazione dei *playground*. Di fatto, le attività sono

quindi risultate "minimamente guidate" [38] e quindi troppo difficili, specialmente per gli studenti più deboli. Nella seconda iterazione, questo effetto è stato in parte mitigato. Se alcuni studenti hanno ancora trovato difficili le attività con Snap!, tutti le hanno largamente apprezzate, suggerendo che esse abbiano rappresentato il giusto livello di sfida: non banali ma neanche eccessivamente difficili, e quindi coinvolgenti senza essere scoraggianti.

Per i crittosistemi più semplici, i *playground* Snap! sono stati usati per far sperimentare gli studenti prima di discutere e analizzare quei sistemi. Le domande-guida sono state essenziali per far sí che gli studenti si concentrassero proprio su quegli aspetti (e quelle limitazioni) che si voleva evidenziare. Solo dopo queste esperienze concrete, le discussioni collettive sono state usate per esplicitare e formalizzare i contenuti più rilevanti. Questo approccio è stato adottato anche con l'attività sul protocollo di Diffie-Hellman.

I sistemi a chiave pubblica, essendo concettualmente più complessi, meno si prestano ad un'impostazione "costruttiva". Tuttavia, anche per questa parte più "frontale", la formalizzazione degli schemi è avvenuta solo dopo aver discusso gli schemi che gli studenti hanno proposto intuitivamente dopo che gli ingredienti essenziali (cioè la coppia di chiavi) erano stati presentati.

In coerenza con gli obiettivi del corso, la valutazione finale [24] si è concentrata sulle idee fondamentali della crittografia. I risultati sono stati molto buoni in entrambe le iterazioni (vedi 4.1): i contenuti principali sono stati certamente colti dagli studenti. Siamo soddisfatti, in relazione sia all'obiettivo di cittadinanza (ogni cittadino dovrebbe avere gli strumenti per capire la società digitale di oggi), sia all'opportunità di orientamento universitario e professionale verso l'informatica e la matematica. Il questionario di gradimento ha confermato il raggiungimento di questi due obiettivi, indicando che gli studenti hanno capito meglio il ruolo della crittografia nella società, hanno circoscritto con più precisione quello di cui questa disciplina si occupa, e hanno visto aumentare il loro interesse per essa.

5.2. Insegnamento della programmazione

I nostri *playground* possono essere visti come linguaggi di programmazione *task-specific* (vedi 2.5). Non hanno, nè possono avere, l'obiettivo di insegnare a programmare. Tuttavia, possono aiutare ad acquisire alcuni principi generali della programmazione, come il fatto che "i programmi sono composti da elementi di base, e diversi modi di comporre questi elementi possono talvolta avere lo stesso risultato, e anche che il programma determina il comportamento del computer (non è magia)" [43, p. 186].

Rispetto ad altri linguaggi di programmazione *task-specific* [43], le nostre attività mostrano più esplicitamente alcuni concetti classici della programmazione (ad esempio, sequenza, composizione di funzioni, variabili, liste). In prospettiva, si potrebbero sviluppare altri blocchi *ad hoc*, insieme ad attività che richiedano un maggiore uso della programmazione (per esempio, che usino altri elementi fondamentali della programmazione strutturata come le strutture condizionali e i cicli; oppure che costringano gli studenti a "guardare dentro" i blocchi forniti per comprenderli e riadattarli).

Allo stato attuale, ispezionare il codice dei blocchi nei nostri *playground* non ha il valore educativo che potrebbe avere. Gli studenti curiosi troverebbero, oltre a molti blocchi predefiniti di Snap!, anche codice Javascript (che abbiamo usato per superare le attuali limitazioni di Snap!). Stiamo progettando una vera e propria gerarchia di "macchine concettuali" (*notional machines*, vedi [31]) a diversi livelli di astrazione, in modo che gli studenti possano vedere progressivamente più dettagli ispezionando i blocchi, scendendo in questo modo i livelli della gerarchia di astrazione, senza però essere sopraffatti da tutta la complessità in un colpo solo.

5.3. Suggerimenti per replicare l'esperienza

Tutti i contenuti (vedi 2.4), il percorso didattico (vedi 2.3), gli strumenti creati e utilizzati nel nostro corso (per esempio i *playground* Snap!) e i materiali (per esempio, il testo per la valutazione finale) sono disponibili con licenza libera [24].

Quanto "guidare" le attività o quanto invece lasciarle alla libera sperimentazione degli studenti puó ovviamente essere stabilito caso per caso. Se il corso è in presenza, i docenti possono avere un quadro più chiaro delle difficoltà degli studenti e affrontarle immediatamente, pur lasciando un alto grado di libertà. Se il corso si tiene a distanza, suggeriamo controlli e riallineamenti più frequenti per poter offrire agli studenti un supporto adeguato.

Anche se il corso ha come obiettivi di apprendimento i principi della crittografia, è chiaro che non si possono comprendere le idee fondamentali di una disciplina senza affrontarle in scenari o sistemi concreti, seppur semplici. Se fosse possibile riservare al corso più tempo di quello da noi impiegato, suggeriamo di dedicarne ancora di piùR alle esplorazioni e alle discussioni, piuttosto che affrontare nuovi crittosistemi (a meno che questi non siano strategici per altri importanti *principi* che i docenti intendano insegnare).

References

[1] ANANE R. and ALSHAMMARI M.T., *A Dynamic Visualisation of the DES Algorithm and a Multi-Faceted Evaluation of Its Educational Value*, in *Proceedings of the 25th ACM Conference on Innovation & Technology in Computer Science Education*, ITiCSE '20, 370–376, ACM, New York, NY, USA (2020), ISBN 9781450368742, URL http://dx.doi.org/10.1145/3341525.3387386.

[2] BELL T., ALEXANDER J., FREEMAN I. and GRIMLEY M., *Computer Science Unplugged: school students doing real computing without computers*, New Zealand Journal of Applied Computing and Information Technology **13** 1, (2009), 20–29, ISSN 1174-0175.

[3] BELL T., THIMBLEBY H., FELLOWS M., WITTEN I., KOBLITZ N. and POWELL M., *Explaining cryptographic systems*, Computers & Education **40** 3, (2003), 199–215, ISSN 0360-1315, URL http://dx.doi.org/https://doi.org/10.1016/S0360-1315(02)00102-1.

[4] 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. and STANDARD S., *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, 303–308, ACM, New York, NY, USA (2012), ISBN 9781450312462, URL http://dx.doi.org/10.1145/2325296.2325367.

[5] BUCHELE S.F., *Two Models of a Cryptography and Computer Security Class in a Liberal Arts Context*, in *Proceedings of the 44th ACM Technical Symposium on Computer Science Education*, SIGCSE '13, 543–548, ACM, New York, NY, USA (2013), ISBN 9781450318686, URL http://dx.doi.org/10.1145/2445196.2445360.

[6] CENTRE. E.C.J.R., *DigComp 2.1: the digital competence framework for citizens with eight proficiency levels and examples of use.*, Publications Office (2017), URL http://dx.doi.org/10.2760/38842.

[7] CS UNPLUGGED, *Principles*, URL https://csunplugged.org/en/principles/.

[8] CSTA, *CSTA K-12 Computer Science Standards, rev. 2017*, Tech. rep., Computer Science Teachers Association (2017), URL http://www.csteachers.org/standards.

[9] DEPARTMENT OF EDUCATION, *National curriculum in England: computing programmes of study* (2013), URL https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study.

[10] DESHPANDE P., LEE C.B. and AHMED I., *Evaluation of Peer Instruction for Cybersecurity Education*, in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, SIGCSE '19, 720–725, ACM, New York, NY, USA (2019), ISBN 9781450358903, URL http://dx.doi.org/10.1145/3287324.3287403.

[11] FEES R.E., DA ROSA J.A., DURKIN S.S., MURRAY M.M. and MORAN A.L., *Unplugged cybersecurity: An approach for bringing computer science into the classroom*, International Journal of Computer Science Education in Schools **2** 1, (2018), 3–13, URL http://dx.doi.org/10.21585/ijcses.v2i1.21.

[12] GRAMM A., HORNUNG M. and WITTEN H., *Email for You (Only?): Design and Implementation of a Context-Based Learning Process on Internetworking and Cryptography*, in *Proceedings of the 7th Workshop in Primary and Secondary Computing Education*, WiPSCE '12, 116–124, ACM, New York, NY, USA (2012), ISBN 9781450317870, URL http://dx.doi.org/10.1145/2481449.2481477.

[13] GREENLAW R., BROWN C., DANNELLY Z., PHILLIPS A. and STANDARD S., *Using a Message Board as a Teaching Tool in an Introductory Cyber-Security Course*, in *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, SIGCSE '15, 308–313, ACM, New York, NY, USA (2015), ISBN 9781450329668, URL http://dx.doi.org/10.1145/2676723.2677221.

[14] GUZDIAL M., *Helping social studies teachers to teach data literacy with Teaspoon languages* (2021), URL https://computinged.wordpress.com/2021/12/22/helping-social-studies-teachers-to-teach-data-literacy-with-teaspoon-languages/.

[15] GUZDIAL M. and NAIMIPOUR B., *Task-Specific Programming Languages for Promoting Computing Integration: A Precalculus Example*, in *Proceedings of the 19th Koli Calling International Conference on Computing Education Research*, Koli Calling '19, ACM, New York, NY, USA (2019), ISBN 9781450377157, URL http://dx.doi.org/10.1145/3364510.3364532.

[16] HSIN W.J., *Teaching Cryptography to Undergraduate Students in Small Liberal Art Schools*, in *Proceedings of the 2nd Annual Conference on Information Security Curriculum Development*, InfoSecCD '05, 38–42, ACM, New York, NY, USA (2005), ISBN 1595932615, URL http://dx.doi.org/10.1145/1107622.1107632.

[17] JOINT TASK FORCE ON CYBERSECURITY EDUCATION, *Cybersecurity Curricula 2017: Curriculum Guidelines for Post-Secondary Degree Programs in Cybersecurity*, ACM, New York, NY, USA (2018), ISBN 9781450389198, URL https://dl.acm.org/doi/book/10.1145/3184594.

[18] K-12 CS FRAMEWORK, *K–12 Computer Science Framework*, Tech. rep. (2016), URL http://www.k12cs.org.

[19] KONAK A., *A cyber security discovery program: Hands-on cryptography*, in *2014 IEEE Integrated STEM Education Conference*, 1–4 (2014), URL http://dx.doi.org/10.1109/ISECon.2014.6891029.

[20] KONAK A., *Experiential Learning Builds Cybersecurity Self-Efficacy in K-12 Students*, Journal of Cybersecurity Education, Research and Practice **2018** 1, ISSN 2472-2707, URL https://digitalcommons.kennesaw.edu/jcerp/vol2018/iss1/6.

[21] LÈDECZI A., MARÓTI M., ZARE H., YETT B., HUTCHINS N., BROLL B., VÖLGYESI P., SMITH M.B., DARRAH T., METELKO M., KOUTSOUKOS X. and BISWAS G., *Teaching Cybersecurity with Networked Robots*, in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, SIGCSE '19, 885–891, ACM, New York, NY, USA (2019), ISBN 9781450358903, URL http://dx.doi.org/10.1145/3287324.3287450.

[22] URL https://www.liceomatematico.it/.

[23] LINDMEIER A. and MÜHLING A., *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 (2020), ISBN 9781450387590, URL http://dx.doi.org/10.1145/3421590.3421630.

[24] LODI M., SBARAGLIA M. and MARTINI S., *Crittografia a blocchi al liceo* (2021), URL https://bigideascryptok12.bitbucket.io/ita.html.

[25] LODI M., SBARAGLIA M. and MARTINI S., *Cryptography in Grade 10: Core Ideas with Snap! and Unplugged*, in *Proceedings of the 27th ACM Conference on Innovation and Technology in Computer Science Education Vol 1*, ITiCSE '22, ACM, New York, NY, USA (2022), ISBN 978-1-4503-9201-3/22/07, URL http://dx.doi.org/10.1145/3502718.3524767, To appear.

[26] LOUI M.C. and BORREGO M., *Engineering Education Research*, in *The Cambridge Handbook of Computing Education Research*, 292–322, Cambridge University Press (2019), URL http://dx.doi.org/10.1017/9781108654555.012.

[27] MA J., TAO J., MAYO J., SHENE C.K., KERANEN M. and WANG C., *AESvisual: A Visualization Tool for the AES Cipher*, in *Proceedings of the 21st ACM Conference on Innovation & Technology in Computer Science Education*, ITiCSE '16, 230–235, ACM, New York, NY, USA (2016), ISBN 9781450342315, URL http://dx.doi.org/10.1145/2899415.2899425.

[28] MCANDREW A., *Teaching Cryptography with Open-Source Software*, SIGCSE Bull. **40** 1, (2008), 325–329, ISSN 0097-8418, URL http://dx.doi.org/10.1145/1352322.1352247.

[29] MIUR, *Regolamento Licei del 16/02/2010*, URL https://archivio.pubblica.istruzione.it/riforma_superiori/nuovesuperiori/doc/Regolamento_licei_definitivo_16.02.2010.pdf.

[30] PAPERT S., *Mindstorms: Children, Computers, and Powerful Ideas*, Basic Books, Inc., New York, NY, USA (1980), ISBN 0-465-04627-4.

[31] SBARAGLIA M., *A Necessity-Driven Learning Design for Computer Science*, in *Proceedings of the 26th ACM Conference on Innovation & Technology in Computer Science Education V. 2*, ITiCSE '21, 664–665, ACM, New York, NY, USA (2021), ISBN 9781450383974, URL http://dx.doi.org/10.1145/3456565.3460017.

[32] SBARAGLIA M., LODI M. and MARTINI S., *A Necessity-Driven Ride on the Abstraction Rollercoaster of CS1 Programming*, Informatics in Education **20** 4, (2021), 641–682, ISSN 1648-5831, URL http://dx.doi.org/10.15388/infedu.2021.28.

[33] SCHWEITZER D. and BOLENG J., *Designing Web Labs for Teaching Security Concepts*, J. Comput. Sci. Coll. **25** 2, (2009), 39–45, ISSN 1937-4771.

[34] SCHWEITZER D. and BROWN W., *Using Visualization to Teach Security*, Journal of Computing Sciences in Colleges **24** 5, (2009), 143–150, ISSN 1937-4771.

[35] SIMMS X. and CHI H., *Enhancing Cryptography Education via Visualization Tools*, in *Proceedings of the 49th Annual Southeast Regional Conference*, ACM-SE '11, 344–345, ACM, New York, NY, USA (2011), ISBN 9781450306867, URL http://dx.doi.org/10.1145/2016039.2016139.

[36] SOMMERS J., *Educating the next Generation of Spammers*, in *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, SIGCSE '10, 117–121, ACM, New York, NY, USA (2010), ISBN 9781450300063, URL http://dx.doi.org/10.1145/1734263.1734302.

[37] TABER K.S., *Constructivism as educational theory: Contingency in learning, and optimally guided instruction*, in H. JALEH, ed., *Educational theory*, 39–61, Nova, New York, NY, USA (2012).

[38] TOBIAS S. and DUFFY T.M., eds., *Constructivist instruction: Success or failure?*, Routledge (2009).

[39] TURNER C.F., TAYLOR B. and KAZA S., *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, 15–20, ACM, New York, NY, USA (2011), ISBN 9781450305006, URL http://dx.doi.org/10.1145/1953163.1953174.

[40] VAN DER LINDEN D., RASHID A., WILLIAMS E. and WARINSCHI B., *Safe Cryptography for All: Towards Visual Metaphor Driven Cryptography Building Blocks*, in *Proceedings of the 1st International Workshop on Security Awareness from Design to Deployment*, SEAD '18, 41–44, ACM, New York, NY, USA (2018), ISBN 9781450357272, URL http://dx.doi.org/10.1145/3194707.3194709.

[41] ŠVÀBENSKÝ V., VYKOPAL J. and ČELEDA P., *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, 2–8, ACM, New York, NY, USA (2020), ISBN 9781450367936, URL http://dx.doi.org/10.1145/3328778.3366816.

[42] WIKIPEDIA CONTRIBUTORS, *Diffie–Hellman key exchange — Wikipedia, The Free Encyclopedia* (2021), URL https://en.wikipedia.org/w/index.php?title=Diffie%E2%80%93Hellman_key_exchange&oldid=1038627636.

[43] YADAV A. and BERTHELSEN U.D., *Computational Thinking in Education*, Routledge (2021), URL http://dx.doi.org/10.4324/9781003102991.

[44] YETT B., HUTCHINS N., STEIN G., ZARE H., SNYDER C., BISWAS G., METELKO M. and LÈDECZI A., *A Hands-On Cybersecurity Curriculum Using a Robotics Platform*, in *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, SIGCSE '20, 1040–1046, ACM, New York, NY, USA (2020), ISBN 9781450367936, URL http://dx.doi.org/10.1145/3328778.3366878.

[45] ZINKUS M., CURRY O., MOORE M., PETERSON Z. and WOOD Z.J., *Fakesbook: A Social Networking Platform for Teaching Security and Privacy Concepts to Secondary School Students*, in *Proc. of the 50th ACM Technical Symposium on Computer Science Education*, SIGCSE '19, 892–898, ACM, New York, NY, USA (2019), ISBN 9781450358903, URL http://dx.doi.org/10.1145/3287324.3287486.

AMS Subject Classification: 97P40, 97P20, 68P25, 94A60

Michael Lodi, Marco Sbaraglia, Simone Martini,
Dipartimento di Informatica-Scienza e Ingegneria, Alma Mater Studiorum-Università di Bologna
Mura Anteo Zamboni, 7, 40126 Bologna, ITALY
e-mail: michael.lodi@unibo.it, marco.sbaraglia@unibo.it, simone.martini@unibo.it

Lavoro pervenuto in redazione il 15.04.2022.