A Study of Effectiveness and Problem Solving on Security Concepts with Model-Eliciting Activities

Jeong Yang
Dept. of Computing & Cyber Security
Texas A&M University-San Antonio
San Antonio, TX, USA
jeong.yang@tamusa.edu

Young Rae Kim
Dept. of Curriculum & Instruction
Texas A&M University-San Antonio
San Antonio, TX, USA
young.kim@tamusa.edu

Brandon Earwood
Dept. of Computing & Cyber Security
Texas A&M University-San Antonio
San Antonio, TX, USA
brandon.earwood@tamusa.edu

***Abstract*—Security is a critical aspect in the process of designing, developing, and testing software systems. Due to the increasing need for security-related skills within software systems, there is a growing demand for these skills to be taught in computer science. A series of security modules was developed not only to meet the demand but also to assess the impact of these modules on teaching critical cyber security topics in computer science courses. This full paper in the innovative practice category presents the outcomes of six security modules in a freshman-level course at two institutions. The study adopts a Model-Eliciting Activity (MEA) as a project for students to demonstrate an understanding of the security concepts. Two experimental studies were conducted: 1) Teaching effectiveness of implementing cyber security modules and MEA project, 2) Students' experiences in conceptual modeling tasks in problem-solving. In measuring the effectiveness of teaching security concepts with the MEA project, students' performance, attitudes, and interests as well as the instructor's effectiveness were assessed. For the conceptual modeling tasks in problem-solving, the results of student outcomes were analyzed. After implementing the security modules with the MEA project, students showed a great understanding of cyber security concepts and an increased interest in broader computer science concepts. The instructor's beliefs about teaching, learning, and assessment shifted from teacher-centered to student-centered during their experience with the security modules and MEA project. Although 64.29% of students' solutions do not seem suitable for real-world implementation, 76.9% of the developed solutions showed a sufficient degree of creativity.**

Keywords—cybersecurity education, computer science education, model-electing activity, MEA, secure programming, CS1, module

I. INTRODUCTION

Software impacts people's lives in a myriad of ways and cyber security affects every computing component in a software system. Cyber security as a discipline has continuously evolved to uncover cyber threats and attacks. These attacks have substantially increased over the past. As security becomes a critical aspect in the design, development, and testing of software systems, it is essential to guarantee that software is safe and behaves as intended. Markettos et al claimed that security must be considered from the ground up in order to build complex hardware and software systems for the new course of vulnerabilities [1]. Saydjari emphasized that software engineers must be responsible for designing and building safe and secure systems and they can do it in conjunction with system risk analysis and management [2, 3]. The study stressed that careless software design and implementation can lead to vulnerabilities and attacks on the application, thus security must be considered throughout the entire software development process. Toward secure software assurance, it is encouraged that security concepts must be taught to beginning programmers [4, 5]. This can be exercised through defensive programming and, secure coding, and secure software development practices [5, 6]. The application of secure coding practices can contribute to quality software systems that are safe and reliable.

While there have been efforts to provide secure coding guidelines [7, 8, 9, 10], not many colleges and universities provide secure coding practices in their programming courses. In many universities, cyber security is taught as an "add-on" track or concentration. Cyber security is so critical that the concepts and skills can no longer be covered as a single topic or in a track. With cyber-attacks and vulnerabilities substantially increased over the past years in terms of frequency and severity, it is important to design and build secure software applications from the ground up. Therefore, it is important to guide the fundamental concepts of secure and defensive programming from the freshman year. The concepts learned in the foundation courses can be applied to build reliable software applications, which can be further integrated with secure software paradigms.

Cyber security modules were developed to meet the demand and assess the impact of these modules on teaching cyber security topics [11]. The goal of the development is to teach cyber security concepts in various Computer Science (CS) courses from the first introductory course to senior-level courses. This paper presents the outcomes of teaching six security modules in a Freshman level course. A set of five modules presented in lectures as well as a sixth module emphasizing encryption and decryption was used as the semester project for the course. Each module is a collection of concepts related to cyber security. The individual cyber security concepts are presented with a general description of a security issue, a sample code with the security issue written in the Java programming language, and a second version of the code with an effective solution.

II. RELATED WORK

A. Cyber Security Modules

The objective of the developed security modules is to keep the modules independent so that they can be easily integrated into the courses. Each module package consists of instructions, lab exercises with solutions, and assessment methods [12]. The first

TABLE I. INCORPORATION OF CYBER SECURITY MODULES IN CS 1

Chapter to Cover Module	Module#.Lesson#	NICE SAs & KSAs	CWE
Ch. 2. Java Fundamentals	1 Integer Errors	T0176, K0070, T0111	CWE-192
	4.1 Secure Variable Declarations	T0686, K0009	CWE-456, CWE-493
Ch. 3. Decision Structures	2 Securing Integer Boundaries & Prevent Overflow	T0176, K0070	CWE-190
	5.1 Secure Division	K0005, T0111	CWE-136, CWE-681
	5.2 Precision		
Ch. 4. Loops and Files	3.1 Floating Point Inputs	T0047, T0728, T083	CWE-20
	3.2 Type Conversion		
	4.2 Scope of Variables	T0686, K0009	CWE-456, CWE-493
Ch. 4. Loops and Files, Ch. 5. Methods Ch. 7. Arrays and ArrayList Class	6 Caesar Cipher – Encryption and Decryption: MEA Semester Project	K0308	CWE-1013

six modules were designed to introduce fundamental security concepts of defensive programming in beginner-level programming courses [12, 13]. The modules are currently available at the NSA's CLAKR Cybersecurity Library for public access [11]. TABLE I presents the six modules along with Chapters to cover the modules with more details in a paper [23].

B. Models and Modeling Perspectives (MMPs) on Learning

The study adopts a Model-Eliciting Activity (MEA) as a project for students to demonstrate an understanding of the security concepts. MEAs are modeling activities designed based on the Models and Modeling Perspectives (MMPs) on learning and problem-solving. The MMPs draw on continuous lineages from Piaget, Vygotsky, and American Pragmatists such as Charles Sanders Peirce and John Dewey, incorporating constructivist views of learning [14]. In the MMPs, learning occurs in model development through engaging in modeling activities such as MEAs, in which students express, test, and revise their solutions (models) for realistic problems in situated contexts and apply their models to different problem situations [15]. Research shows that model development in MEAs involves improving conceptual understanding [16]. In addition, realistic problem contexts make it easier for students to associate their knowledge and skills required for problem-solving tasks. As a result, the learning modules with the MEA project are expected to positively affect the instructor's effectiveness and the student's attitudes and interest as well as their learning experience in problem-solving [9, 10, 27].

C. Model-Eliciting Activities (MEA)

Model-Eliciting Activities (MEAs) are open-ended, problem-solving activities in which groups of three to four students work to solve complex problems in a classroom setting [15]. One of the important differences between MEAs versus typical engineering problem-solving activities is the emphasis on multiple iterations of expressing, building, testing, and revising conceptual models [18].

During MEAs, students are required to develop or design mathematical/scientific/engineering tools or artifacts that an imaginary client needs to solve a realistic problem [15, 19]. Student groups are given an article or video as an advanced organizer, introducing the realistic context and providing background information. After that, students individually answer readiness questions making them familiar with the practical context, and ready to engage in the problem task. A problem statement is provided for the students that may specify the client's requirements. Students work in small groups of three to four to develop alternative solutions and choose the best one. They design and build it as a prototype. Then they test and revise it to meet the needs of their client successfully. Finally, student groups present their solutions and ideas to the class, and they are given time for self-reflection and final revision of their models.

MEAs have been proven as an effective method to help engineering students become better problem solvers [16, 20, 21]. A key feature of MEAs which makes them very suitable for this study is that MEAs are meant to be complementary materials for a curriculum, with the result that they can easily be integrated into existing curricula [19]. MEAs also have the potential of providing students with experiential learning opportunities, on engaging projects in the domain in which they are implemented [22] – computing and cyber security – for this study. MEA also helps students in becoming better problem solvers [16, 20, 21].

III. IMPLEMENTATION OF SECURITY MODULES WITH MEA

A. Incorporation of Cyber Security Modules

During the fall semester of 2019, the nine lessons (six modules) were taught in CS 1 courses at two institutions: Texas A&M University-San Antonio (SA) and San Antonio College (SAC). The book used for the course was Starting Out with Java: From Control Structures through Objects, 7th Edition. TABLE I outlines how the concepts of the modules and lessons were integrated with chapter materials, how they were related to CWE (Common Weakness Enumeration) [29], and their Specialty Areas (SAs) and Knowledge, Skills, and Abilities (KSAs) in the NICE category [30]. Incorporating each of the concepts in these modules into lectures depends on both the topic covered and the approach to resolving the related security issue.

Two sections of the CS 1 course at each institution were used as a control group and a treatment group. The treatment group included the security modules in lectures with the MEA while the control group did not. To measure the overall effectiveness of teaching security modules and MEA, both the instructors' effectiveness and the students' attitudes and interest were measured. Sstudents in the treatment group were first introduced to computing concepts related to both the security issue and the appropriate solution. This was followed up with a continual review that requires exploring use-cases for the programming mechanisms presented as solutions to the security issues discussed. In addition to the security modules presented in lectures, students were also given a hands-on approach to understanding the concepts through a Model-Eliciting Activity (MEA). The semester MEA project related to encryption and decryption was implemented into the course as an MEA.

TABLE II. PRINCIPLES FOR GUIDING MEA DEVELOPMENT WITH CYBER SECURITY EXAMPLES

Principle	Description	Principle Description Example
Reality	Requires the activity to be posed in a realistic everyday context.	Student teams develop an encryption system to protect data that could be stolen – ex: passwords.
Model Construction	Requires the development of an explicit description, explanation, or procedure for a significant system, constructed, modified, & refined.	Students specify requirements, develop a pseudocode algorithm and implement the algorithm. Implementation must map to the concepts of secure coding rules, and topics covered from security modules. Solution is refined cyclically.
Model Documentation	Requires to create some documentation to their solution and process to the problem situation.	Student teams prepare documentation (report, presentation, demonstration) that contains key concepts of problem-solving strategies in their algorithm.
Self-Assessment	Contains criteria the students can identify and use to test and revise their current ways of thinking.	Student teams test if their developed system meets the requirements and identify any issues with customer-driven test cases. They revisit the algorithm and revise it based on the test results.
Generalizability	Requires students to produce solutions that are reusable with others and modifiable for other closely related engineering situations.	Students' algorithms should be reusable for developing of other encryption systems. A final product should allow others to reuse the product for regularly updating their encryption system in the future.
Effective Prototype	The model produced will be as simple as possible, yet still significant for learning purposes.	MEA aims for developing technological literacy in computer security, especially the concept of encryption methods.

TABLE III. ENCRYPTION AND DECRYPTION OF THREE CIPHER ALGORITHMS USED IN INDIVIDUAL ACTIVITY

Chapter	Encryption	Decryption
Caesar Cipher	Shift each letter three letters forward in the alphabet (sending the last three letters of the alphabet to the first three), process it mathematically: first replace each letter by an element of Z26 (0-25), a function f: **f(p) = (p + 3) mod 26**, p represents a letter.	Use an inverse function of f, is used: $\mathbf{f^{-1}(p) = (p - 3) \bmod 26}$, the letter represented by p, is replaced with the letter represented by (p - 3) mod 26: use ((p + 26) – 3) mod 26 when p = 0, 1, 2.
Affine Cipher	Caesar's shift cipher can be generalized to enhance security by using a function: **f(p) = (ap + b) mod 26,** where a and b are integers.	To recover the cipher, the inverse function of f, is used: $\mathbf{f^{-1}(p) = (p-b)/\ a}$ mod 26. If (p-b) does not divides a, use ((p-b) + 26 *i) / a.
Block Cipher	Split letters into blocks of size m. (If the number of letters in the messages is not divisible by m, add some random letters at the end to fill out the final block.) $c1c2...cm = p\sigma(1)p\sigma(2)....,p\sigma(m$.	To decrypt a cipher text block *c1c2...cm,* transpose its letters using σ^{-1}, the inverse of the permutation σ.

B. Development of MEA Project

The development of MEAs is designed by six design principles: Reality, Model Construction, Model Documentation, Self-Assessment, Generalizability, and Effective Prototype [18]. TABLE II provides a description of each of the principles that map to a real example for this study.

a) MEA Individual Activity: MEA implemented for the study involved three simple encryption. To introduce the realistic context and provide background information, students were given an article that describes the background of the Caesar Cipher. They also learned about the Caesar cipher, affine cipher, and block cipher along with their encryption and decryption formulation and examples, and then answered a set of readiness questions related to each algorithm to demonstrate their understanding (TABLE III).

b) MEA Group Activity: To follow up on the individual activity, student groups were given the task of designing a unique cipher algorithm based on the principles learned from the individual activity [14, 23]. Students were presented some background information about a problem requiring the design of a new encryption algorithm. They were expected to use the knowledge gained from the individual activity to design an entirely new algorithm [23]. Student groups prepared both a written description, either as pseudocode or step by step instructions, of their algorithm as well as a visual description, either as a diagram or flowchart. On the following lecture day, groups presented their solutions to the rest of the class.

IV. EXPERIMENTALENTL DESIGN AND METHODOLOGY

Two experimental studies were conducted on teaching effectiveness of cyber security modules with MEA and sstudents' experiences in conceptual modeling tasks in problem-solving, which was adapted from MEAs and MMPs on learning and problem solving described in section II.

A. Research Questions

a) Effective Study: To study the teaching effectiveness of cyber security modules with MEA, the nature of the intervention is investigated by using the design experiment methodology [24]. This methodology investigates how a particular intervention affects student learning and instructor teaching practices [25]. This study has two parts on students' attitudes and interests, and instructor effectiveness: 1) *Can the implementation of cyber security modules through MEAs change students' attitudes and interest in learning computer science? 2) To what extent do instructors change their attitudes towards student learning and their teaching practices because of the implementation of cyber security modules through MEAs?*

b) Study of Problem Solving: To study students' conceptual modeling tasks in problem-solving, the results of student outcomes through the MEA are studied with three questions*: 3) Whether students conceptually connected with the project along with course contents, if not, what misconceptions did the students have? 4) Are the solution and ideas applicable to the implementation of real-world applications? 5) Are developed solutions creative?*

TABLE IV. OPEN-RESPONSE SURVEY ON STUDENT LEARNING EXPERIENCE

Question	Involved Groups
1. How likely are you to enroll in the [Next Course in the Computer Science Sequence] next semester?	Both
2. Explain briefly what helps you learn in the Computer Science courses at your institution, preferably by using an example.	Both
3. What changes, if any, would you suggest to make the courses more helpful?	Both
4. Have you become more competent due to participation in the courses?	Both
5. Do the cyber security modules and MEAs contribute to your interest and understanding of computer science?	Treatment Group

TABLE V. PRE- AND POST-BELIEFS INTERVIEW PROTOCOL QUESTIONS

Pre-Interview		Post-Interview Question
Question	Category	
1. How do you describe your role as the instructor?	Teaching practice	1. What are some changes in your classrooms after the use of MEAs for cyber security modules? 2. What are some differences between your expectation and your observation in the student work through the use of MEAs for cyber security modules?
2. How do your students best learn engineering?	Student learning	
3. How do you maximize student learning in your classroom?	Teaching practice	
4. How do you know when your students understand?	Assessment	
5. How do you decide what to teach or what not to teach?	Teaching practice	
6. How do you decide when to move on to a new topic in your class?	Assessment	
7. How do you know when learning is occurring in your classroom?	Student learning	

B. Participants and Procedure

52 undergraduates and one instructor at SA and 22 undergraduates and one instructor at SAC participated in this study. Their participation was voluntary. The instructors taught two sections of the course: one with the implementation of the cyber security modules with MEA (treatment group: 26 students at SA and 12 students at SAC) and another without the implementation (control group: 26 students at SA and 10 students at SAC). As students were enrolled in different sections of the same course, participants were considered to be randomly assigned to these groups. For the treatment groups, after the security modules were covered, students were given the individual activity of the MEA as an advanced organizer for the concepts of cipher and encryption algorithms. In following up on the individual activity, student groups were formed and given the MEA group activity of designing a unique cipher algorithm. The groups presented their solutions to the class.

C. Data Collection and Analysis

a) Effective Study: Quantitative and qualitative data were collected through open-response student surveys from both treatment and control groups, and semi-structured pre- (beginning of the semester) and post-interviews (ending of the semester) of the instructor. The open-response student survey included four questions for both treatment and control groups to explore the student learning experience, and an additional question for the treatment group to examine the effectiveness of the module implementation with the MEA (TABLE IV).

The pre-and post-interview protocol for the instructors included seven questions, which were adapted from previous studies [27, 28]. This is to assess instructors' current views on instructional practices, student learning, and student understanding. Additional questions were asked for the post-interview to assess instructors' views on the implementations of the cyber security modules and MEA. For each interview, field notes were taken. TABLE V shows seven pre-interview questions categorized into teaching practice, student learning, and assessment, and two post-interview questions. The observation instrument of instructor implementation of the security modules and MEA consisted of the researchers' field notes and the instructors' interaction with students. The interview field notes, and survey responses were analyzed by both deductive and inductive approaches to coding the qualitative data [29, 30].

First, the two researchers established the coding schemes with a consensus on the codes (categories) to student survey responses to questions 1 and 2 in TABLE IV as they became apparent from the data. The following five Likert Scale of interests was applied as the codes to the student responses to question 1: (1) Not likely, (2) Possibly, (3) Likely, (4) Very Likely, and (5) Definitely. For question 2, the formulated codes were (1) "not sure"; (2) "student-centered" strategies (e.g., hands-on, by doing, collaborative, interactive); (3) "neutral" (e.g., assignments, repetition); and (4) "teacher-centered" strategies (e.g., detailed instructions; PPT slides; textbook).

Second, the instructors' responses to the seven questions in TABLE V were coded by two researchers based on preset rubrics that were adopted from previous studies [27, 28, 31]. The rubrics for each question consist of five categories ranging from more teacher-centered to more student- centered beliefs: (1) Traditional, (2) Instructive, (3) Transitional, (4) Emerging Constructivist, and (5) Experienced Constructivist. The most teacher-centered beliefs were coded (1) for *Traditional*, "which indicates beliefs that teachers are providers of knowledge." The code (2) *Instructive* indicates "beliefs that students should have experiences that mimic the teacher or are closely monitored and directed by the teacher." "Beliefs that instruction should be teacher-led but have student input" were coded (3) for *Transitional*. The codes (4) *Emerging Constructivist* and (5) *Experienced Constructivist* indicate more student-centered beliefs. A graphical representation using asterisks was also adopted from a previous study to explore instructors' shift in overall belief system over the semester [27]. The missing responses from some students to each question were not included in the data analysis process. Thus, the total numbers of student responses were different for each question. In coding the data by the two researchers, Cohen's K coefficient of the inter-rater agreement was 0.91, indicating an acceptable level of reliability [30]. The two researchers also discussed differences in coding and made a consensus on the coding discrepancies.

b) Study of Problem Solving: For the study of exploring students' experiences in the MEA *Cipher Algorithm* involving

TABLE VI. LIKELIHOOD OF TAKING THE NEXT CS COURSE

Institution	Group	Not Likely	Possibly	Likely	Very Likely	Definitely	Total
SA	Treatment	1 (4.8%)	1 (4.8%)	0	14 (66.7%)	5 (23.8%)	21
	Control	3 (13.0%)	1 (4.3%)	4 (17.4%)	12 (52.2%)	3 (13.0%)	23
SAC	Treatment	1 (10%)	2 (20%)	2 (20%)	5 (50%)	0 (0.0%)	10
	Control	0 (0.0%)	1 (9.1%)	2 (18.2%)	5 (45.5%)	3 (27.3%)	11

TABLE VII. CIRCUMSTANCES THAT HELPED STUDENTS LEARN IN THE COMPUTER SCIENCE COURSE

Institution	Group	Not sure	Teacher-centered	Neutral	Student-centered	Total
SA	Treatment	0 (0.0%)	7 (28%)	9 (36%)	9 (36%)	25
	Control	1 (4.8%)	6 (28.6%)	7 (33.3%)	7 (33.3%)	21
SAC	Treatment	0 (0.0%)	1 (12.5%)	1 (12.5%)	6 (75%)	8
	Control	0 (0.0%)	3 (30%)	3 (30%)	4 (40%)	10

conceptual modeling tasks in problem-solving, the outcomes of the MEA project were collected and analyzed from the treatment groups at the end of the semester. The MEA outcomes were student group reports that include their solutions, processes with written and visual descriptions, and group presentations. The MEA outcomes were coded by two faculty in the department of computing and cyber security. The coding was conducted focusing on students' understanding of cyber security concepts, feasibility of their solutions as real-world applications, and creativity of the solutions. Any discrepancies in the coding were then discussed by the researchers, and resolved through consensus.

V. RESULTS AND DISCUSSION

A. Results of Effectiveness Study

a) Student Experience with MEA: The student survey responses to the first question in TABLE IV were coded by the five Likert Scale of interests. The responses to the second question were coded by the four categories: (1) "not sure"; (2) "student-centered" strategies; (3) "neutral"; and (4) "teacher-centered" strategies. The responses for these two questions were explored to indirectly examine the impact of the use of the MEA for the cyber security modules on student interest and understanding of learning computer science, along with the direct question 5 for the treatment group.

TABLE VI summarizes data analysis using the five Likert Scale showing the interest of enrolling in a CS course next semester. Although there is no statistically significant evidence, this finding from the frequency counts and percentages is useful to explore general patterns in the data [28]. For example, a test of independence was calculated comparing the interest of students at SA with the intervention of implementing the cyber security modules and MEA, $\chi^2\,(2, N = 44) = 5.06; p = 0.0798$. Thus, there is no statistically significant association between the intervention and students' interest of enrolling in a CS course next semester. However, at SA, 90.5 % (Very Likely and Definitely: 19/21) of the participants in the treatment group wanted to enroll in the next CS course. Only (2/21) 4.8 % of them said it was unlikely for them to take the next course. In the control group, 65.2% (15/23) of participants wanted to take the next course in the next semester, and 13% (4/23) of them didn't want to take the next course in the CS course. The patterns in the data might be able to indicate the possibility of a positive impact on student interests in CS using MEAs. However, at SAC, the patterns in the data indicate a negative impact of MEAs on student. 50% (Very Likely: 5/10) of the participants in the treatment group wanted to enroll in the next CS course. 20% of them (2/10) said it was unlikely for them to take the next course. In the control group, 72.7% (Very Likely and Definitely: 8/11) of participants wanted to take the next course in the next semester. The difference between the two institutions might be due to the difference in the sample size and student's individual personal situations. For example, at SAC, one of the two students who responded as "Possibly" enrolling in the next CS course in the treatment group indicated his personal situation as follows: "I would like to enroll but because of my financial problems I am not sure to enroll."

TABLE VII shows the results of the second question that is related to circumstances that helped students learn CS concepts. At SA, there was no significant difference between treatment and control groups. The data only shows that many participants in both groups did learn from student-centered environments. Students thought they learned the concepts better in hands-on activities, group activities, or real-world problem solving, which are the main characteristics of MEAs. However, at SAC, there was a difference between treatment and control groups. 75% (Student-centered, 6/10) of the participants in the treatment group, comparing to 40% (4/10) of the participants in the control group, responded that student-centered strategies, such as hands-on and group projects, helped them learn better. This might also be able to indicate that the use of the MEAs on teaching cyber security modules could be a way to enhance students' interest and give them a better understanding.

For the question 5 in TABLE IV, approximately 81% (17/21) of the students in the treatment group at SA and 90% (9/10) of the students in the treatment group at SAC expressed that the use of the cyber security modules and MEA contributed to their interests and understanding of computer science. In addition, at SA, 5 students from the treatment group suggested more hands-on activities, group activities, or real-world problems to make the course more helpful. These responses indirectly reflect their experience with the MEA. Conversely, only one student from the control group suggested more real-world problems. This difference could support the contribution of the implementation of the cyber security modules and MEA to students' interest and understanding of computer science. In summary, students have expressed that the MEA in the corporation of the cyber security modules in the course enhances their interests and attitude toward learning in computer science.

TABLE VIII. INSTRCUTOR BELIEFS OF TEACHING, LEARNING, ASSESSMENT

Instructor		A		B	
Interview		First (beginning of the semester)	Second (end of the semester)	First (beginning of the semester)	Second (end of the semester)
Teaching	Role as Instructor	(1)	(3)	(3)	(5)
	Maximize Student Learning	(2)	(2)	(2)	(3)
	What to Teach	(1)	(1)	(1)	(3)
Learning	How Students Learn Best	(2)	(3)	(3)	(2)
	Learning Occurs	(3)	(3)	(2)	(3)
Assessment	When Students Understand	(3)	(5)	(2)	(2)
	When to Move on	(1)	(2)	(1)	(4)

TABLE IX. INSTRUCTOR CHANGE OF BELIEFS OVER THE SEMESTER

Instructor	Interviews	Traditional (1)	Instructive (2)	Transitional (3)	Emerging (4)	Constructivist (5)
A	1st Interview	***	**	**		
	2nd Interview	*	**	***		*
B	1st Interview	**	***	**		
	2nd Interview		*	****	*	*

* Each asterisk represents the code the answer received for one of the seven interview questions.

b) Instructor Change in Beliefs Over the Semester: TABLE VIII shows the instructors' beliefs on teaching, learning, and assessment before and after the implementation of the cyber security modules and MEA. As already reported [23], instructor A's beliefs on teaching, learning, and assessment shifted from teacher-centered to student-centered with the MEA project.

Instructor B's Beliefs: In the first interview at the beginning of the semester, instructor B revealed that he was an instructor who displayed a combination of "traditional" and "instructive" traits. His decision on what to teach was guided by curriculum: "[I] have to go by the state learning objectives"; "If we can cover everything, then he implements the extra topics" (traditional). His decision on when to move on to a new topic was dependent on his agenda: "[we] can have extra projects if students meet my requirements" (traditional). To maximize students learning, he "gauge[d] how the class is doing" (instructive). He knew when learning was occurring by giving "weekly quizzes" (instructive) and tried to assess student understanding through monitoring how "students are interacting with questions" (instructive). He viewed his role as a teacher "to bring everyone to a higher level of the concepts" (transitional). He believed that students best learn "based off lecture and book, then apply with a hands-on project" (transitional).

After the MEA project, instructor B described his role as "a mentor to students [and] encourage to learn and do more beyond the scope of the classroom" (experienced constructivist). He focused on "meeting at least minimum requirements" and then "go beyond the minimum" when deciding what to teach (transitional). He also believed that he could maximize student learning by providing "multiple techniques, lecture/hands-on, repetition, and experience" (transitional). His emphasis on student feedback is a significant change towards a more student-centered view in his beliefs on assessments. He decided when to move on to a new topic "based on student feedback, he spent another class period to recover misunderstood topics" (emerging constructivist). To assess student understanding, he utilized "quizzes for weekly assessment" and asked his students "a lot of questions in class" (instructive). He still believed that students best learn a "combination of reading/material + application of concepts" (transitional). He knew whether learning was occurring through "[students'] nodding, confirmation after asking questions, taking a poll from the class" (transitional).

TABLE IX represents instructors' shift in overall belief system. A general shift of the instructors' responses to the right is represented in the second interview, comparing to the first interview. Results indicates that both instructors exhibited a shift in their beliefs toward a more student-centered view. The instructors also shifted from an instructor who displayed a combination of "traditional" and "instructive" traits to a more student-centered instructor having "transitional," "emerging constructivist," and "constructivist" views. This indicates that the instructors' beliefs about teaching, learning, and assessment shifted from teacher-centered to student-centered, during their experience with the MEA. This meaningful finding answered the research question 1.

c) The limitations of the study include: 1) The effectiveness study solely relied on student surveys and instructor interviews. 2) The instructor interview data were based on two cases that limit its generalizability.

B. Results of Problem Solving

Overview of MEA Outcomes: For each of the treatment groups involved in the MEA, there were three outcomes: 1) combining the previously learned algorithms, 2) using a combination of previously learned algorithms and additional algorithms not covered in the individual assignment, or 3) attempting to produce entirely new creative algorithms. While this order does highlight the least to most inventive approaches to solving the problem, this order represents the most to least practical ideas. The most inventive ideas presented make for interesting design approaches but would prove to be impractical or infeasible in implementation at the level of students. TABLE X shows the algorithms involved in that group's proposed solution showing how the students applied the learned security modules to their models (algorithms). Overall, out of 13 groups (9 SA and 4 SAC groups), 5 groups utilized Caesar Cipher (1 modified and 4 direct use), 6 groups used Affine Cipher (1 modified and 4 direct use), 4 groups utilized Block Cipher (all direct use), and 4 groups presented other algorithms. 6 groups used the approach of combining two Cipher algorithms.

TABLE X. CIPHER ALGORITHM SOLUTION FOR TREATMENT GROUPS

Group		Solution	Caesar Cipher, Affine Cipher, Block Cipher, Others
SA	1	Affine cipher depending on a character's position in the alphabet as well as a block cipher with blocks of length 5	**Modified Affine Cipher:** First half of the alphabet: f(x)= (x+67) mod 13. Second half of the alphabet: f(x) = (–43) mod 13. **Direct Block Cipher**: Blocks of length 5 with the following ordering {4,5,1,2,3}
	2	Key encryption that applies a different cipher to different segments of the message along with salting	**Different ciphers** to different segments of the message
	3	Affine cipher along with a Vigenere cipher and a double transposition	**Direct Affine Cipher:** 3*character+4 **Vigenère cipher** & double transposition
	4	Caesar cipher with reflection and shifting based on the length of a word	**Modified Caesar Cipher:** (ROT(13)+ [length_of_word])
	5	Block cipher and affine cipher followed by salting with SHA1 algorithm	**Modified Affine Cipher:** [word_length] * (letter – 1) – word_length **Direct Block Cipher:** Blocks of length 3 with the arrangement {2,3,1}
	6	Caesar with character obfuscation by using special characters	**Direct Caesar Cipher:** Shift to special characters
	7	Block cipher with blocks of different lengths	**Direct Block Cipher:** Lengths of 3 and 4 with repeated Z for padding
	8	Block cipher with unique padding and Caesar cipher	**Direct Caesar Cipher**: ROT(-2) after performing the block cipher **Direct Block Cipher by words:** every three letters with extra spaces filled in with the alphabet backward; pos 1 and 3 are swapped
	9	Affine with obfuscation by inserting additional characters at every odd position of the original String	**Direct Affine Cipher:** (3n)
SAC	1	Caesar Cipher	**Direct Caesar Cipher:** (ROT13), Binary obfuscation
	2	Affine Cipher	**Direct Affine Cipher:** (5n+9)
	3	Caesar Cipher	**Direct Caesar Cipher:** (user selects shift)
	4	Enigma(ish) Affine Cipher	**Direct Affine Cipher:** (5n+3) Hard coded switch for alphabet performed before Affine cipher (Enigma[ish])

TABLE XI. ANALYSIS RESULTS OF RESEARCH QUESTION 3

Group		Analysis
SA	1	Reasonable understanding of both the affine and block cipher through a meaningful combination of these two algorithms.
	2	The complexity of the idea suggests a strong understanding of concepts, but it is difficult to confirm any knowledge about specific details since the description of the algorithm is highly abstract.
	3	Reasonable understanding of how to perform encryption.
	4	Students did not seem to realize which type of cipher they were applying (i.e. they applied a Caesar cipher twice).
	5	Strong understanding of how to develop a highly secure algorithm.
	6	Students seem to grasp concepts but may confuse obfuscation with security.
	7	Reasonable understanding of the block cipher.
	8	Good understanding of how to incorporate both the block and Caesar ciphers together.
	9	Good grasp on how to do the cipher. The step by step instructions even indicate an understanding of ASCII values and the use of arrays.
SAC	1	Obfuscation of switching between binary and UTF-8 is probably not helpful.
	2	Understand how to implement encryption and decryption of an Affine Cipher. They did not make this cipher any more or less difficult to break than it has always been.
	3	Presentation and accompanying documentation are vague. Group seems to understand how to encrypt / decrypt a Caesar Cipher.
	4	Possible confusion on the difference between Block / Caesar Cipher and hard coded switching of characters. Claimed Caesar Cipher and Block Cipher, but neither present.

TABLE XII. ANALYSIS RESULTS OF RESEARCH QUESTION 4

Group		Analysis
SA	1	The initial idea is similar to implementations in real world applications, but additional steps are necessary to ensure a high degree of security.
	2	The idea conveys a great deal of security, but an implementation of the idea does not seem feasible.
	3	Each of the methods incorporated is a variation on shifting the individual characters, which is not a difficult process to decrypt.
	4	While using word length reduces the effectiveness of frequency analysis, decrypting a Caesar cipher is an overall trivial process.
	5	Depending on the length of a word, the individual letters that are left as a number can be up to 4 digits long. Unless the message is sent as an array, how will you know where one number stops, and another begins?
	6	The shift to special characters is still the same as any other shift using a Caesar cipher. Decrypting messages is equally easy to do.
	7	The padding helps reveal how the characters are being rearranged, which makes decrypting messages trivial.
	8	The frequency of characters is not changed by using a Caesar cipher with a constant shift, so a frequency analysis can restore the String to its original contents. The block cipher does not require a complex set of rearrangements to solve.
	9	The absence of one letter words (i.e. 'a' or 'I'), as well as the fact that all words are of even length, means that all words are doubled in length. The cryptographer may notice that every other letter is a distinct amount of characters apart which would give away the affine cipher.
SAC	1	If the security questions / answers were used in a way that private keys are used rather than the very slow implementation of human I / O.
	2	This code would be broken fairly easily by cryptologists.
	3	Simple Caesar Cipher.
	4	If multiple different hard coded alphabet switches were created, and hard guidelines were implemented whereas, messages would not be repeated (e.g. the Nazi's ending every message with "Heil Hitler"), then perhaps this could be a hard code to break.

a) MEA Assessment: TABLE XI shows the analysis results regarding students' understanding of cyber security concepts to address the research question 3. Overall, all students appeared to connect with the idea of the project and showed a good understanding of cyber security concepts. Most SA groups (8/9) used combinations of Ciphers, with two groups using

TABLE XIII. ANALYSIS RESULTS OF RESEARCH QUESTION 5

Group		Creative	Analysis
SA	1	Yes	The idea primarily shows an effective modification to the affine cipher that begins to improve upon the security of this algorithm.
	2	Yes	The overall idea suggests a highly complex approach to encryption that goes well beyond what is covered during the course.
	3	Yes	While the affine cipher is not modified in any meaningful way, the addition of other encryption algorithms is a unique solution.
	4	Yes	The consideration for varying the shift of certain characters shows a noticeable difference in the original Caesar cipher.
	5	Yes	Combines two cipher algorithms, makes some noticeable modifications to the affine cipher, and incorporates other concepts.
	6	Yes	The added layer of obfuscation is an interesting direction for the Caesar cipher.
	7	No	This is a simple block cipher that only uses two different lengths for the blocks.
	8	Yes	Combines two of the cipher algorithms & introduces another element of complexity by using different characters to fill in blank.
	9	Yes	The solution expands on the affine cipher with the addition of a naïve salting process.
SAC	1	Yes	Security questions, a text reversal, and switching letters to binary
	2	No	This is simply an Affine Cipher.
	3	No	This is a Caesar Cipher that can have different amount of rotation.
	4	Yes	This is very similar to the Enigma machine, but they designed it to be that way.

methods outside of the three Ciphers discussed (group 3 with a Vigenere Cipher, group 5 with MD5 or SH1). Group 7 appeared to utilize a direct Block Cipher (blocks of two different sizes), while group 6 used a direct Caesar Cipher, with characters instead of integers. All SAC groups seemed to utilize a single instance of a particular cipher. The purpose of some ideas seemed questionable (such as asking user security questions during the encryption/ decryption process, converting to binary), but each group was successful in taking a string of text, modifying it to a process, and demonstrating how to undo it.

TABLE XII shows the analysis results on feasibility of the students' solutions as real-world applications to address the research question 4. Many of the solutions (64.29% (9/14)) were not deemed suitable for real-world implementation and would not be practical at a professional level. This is specifically because the solutions are primarily variations on the Caesar, affine, and block ciphers, which do not provide a sufficient level of security. Thus, that would be a misconception about what constitutes secure data because the students have not been exposed to techniques for breaking these encryption algorithms. Many SAC solutions were direct applications of a single instance of a Cipher, with the most secure (SAC group 4) utilizing a single hard-coded substitution table. SA group 1 did remark during their presentation that Microsoft implements password security similar to their method of splitting the alphabet, but MS Security Documentation describes using additional security for Man-in-the-Middle Attacks, such as "proprietary signaling protocol which leverages TLS 1.2 and AES-256 (in GCM mode) encrypted UDP / TCP channel" [26]. A few of the solutions did provide other techniques, such as salting, that potentially improve security. Although it is difficult to know if SA group 2 should be able to implement their solution due to its high level of abstraction, it would be interesting to further investigate possible implementation options with the specific details.

TABLE XIII shows the results with regard to the creativity of the developed solutions, which also relates to the question 5. While they are at different levels of creativity, the majority of the ideas (76.9%: 10/13) seem to be creative, with the exception of SA group 7, which used a direct block cipher, with alternating block lengths of 3 and 4. SAC groups 1 and 4 developed the most creative solutions at SAC. SAC group 1 included security questions, a text reversal, and switching letters to binary, while SAC 4 included a random hard-coded substitution table. SA groups 3 and 5 used methods that were not included in the project's handout. SA group 2 is particularly creative in its use of a pool of many different algorithms and presented a unique algorithm in that most groups either modified the behavior of a single Algorithm, or blended elements from two.

b) Elaborating Creative Solution: There were also some interesting new ideas and creative algorithms presented. One example is the solution presented by SA group 2. They presented an abstract concept that would utilize many different ciphers (Fig. 1). A preliminary key would be sent from sender to receiver, with the following keys sent in "parcels," that would be unlocked by the preceding key. A complete message would be divided into a random total number of segments, and each segment would occupy its own parcel. Each segment would contain a random total number of possible encryption algorithms (the group says "up to 15"), selected from a pool of total encryption algorithms. From there, one algorithm will be randomly selected, with that segment being encrypted accordingly, and stored in its own parcel.

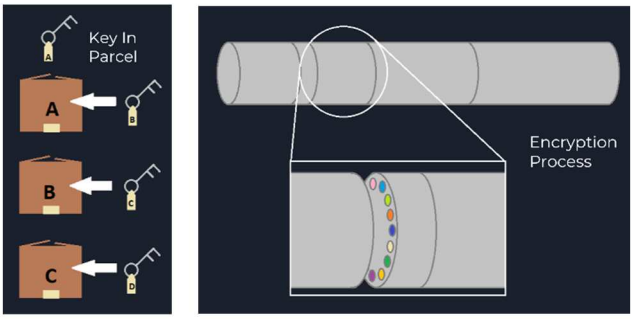

Fig. 1. SA Group 2 Solution - Encryption Process with Key in Parcel and Salting.

This process will continue for all segments, independently, and with all actions saved and recorded as the key. Finally, the resulting message will be salted with garbage values. This approach left certain questions unanswered though, one of which being: in a Man in the Middle Attack, with keys being sent in each parcel, if the attacker un-encrypts one parcel and gains access to that key, how effective would the remainder of the security of the algorithm be? Also, does providing an attacker with numerous parcels (each containing a key) reduce security by making numerous potential access points for the attacker? Other considerations like CPU overhead, the effectiveness and security of the random number generator used, and ultimately how they would implement the algorithm remain.

VI. Conclusion

This paper presents the outcomes of implementing six security modules with an MEA project at two institutions. Two studies were conducted: 1) Teaching effectiveness on students' interests/attitudes and instructor effectiveness, 2) Students' experiences in conceptual modeling tasks in problem-solving. Study results showed student's enhanced interest in learning. Although there is no statistically significant evidence, the difference between two SA groups (90.5% in the treatment group and 65.2% in the control group) in the likelihood of taking the next CS course. This supports a positive impact on student interests. At SAC, no positive impact was found. Approximately 81% of the students in the treatment group at SA and 90% of the students in the treatment group at SAC expressed that the use of the cyber security modules and MEA contributed to their interests and understanding of CS concepts.

For instructor effectiveness, results indicate that both instructors exhibited a shift in their beliefs towards a more student-centered view from a teacher-centered view during their experience with the cyber security modules and MEA. The instructors also shifted from an instructor who displayed a combination of "traditional" and "instructive" traits to a more student-centered instructor having "transitional," "emerging constructivist," and "constructivist" views. The instructors' responses also support a positive impact of the use of MEAs on their beliefs and decisions on teaching, learning, & assessment. On the problem-solving strategies from the treatment groups, all students appeared to connect with the idea of the project and showed a good understanding of cyber security concepts. 64.29% of the solutions were not deemed suitable for real-world implementation and may not be practical at a professional level. The variations on the three ciphers do not provide a sufficient level of security. However, 76.9% of the developed solutions seem to be creative at differing levels of creativity.

Acknowledgment

This material is based upon work supported by the National Science Foundation (NSF)'s Grant No #1832433.

References

[1] Markettos, A. T., Watson, R. N. M., Moore, S. W., Sewell, P., & Neumann, P. G. (2019). Through Computer Architecture, Darkly, Communications of the ACM, Vol. 62 No. 6, Pages 25-27, 10.1145/332528.

[2] Saydjari, O. Sami. (2019). Engineering Trustworthy Systems: A Principled Approach to Cybersecurity. Communications of the ACM, Vol. 62 No. 6, Pages 63-69, 10.1145/3282487.

[3] Stamat, M. L. & Humphries, J. W. (2009). Training ≠ Educating Secure Software Engineering Back in the Classroom. WCCCE '09 May 1-2, 2009, Burnaby, BC, Canada. ACM 978-1-60558-415-7.

[4] Yang, J., Lodgher, A., & Lee, Y. (2018). Secure Modules for Undergraduate Software Engineering Courses. 2018 IEEE Frontiers in Education Conference (FIE), doi: 10.1109/FIE.2018.8658433.

[5] Yang, J. & Lodgher, A. (2019). Fundamental Defensive Programming Practices with Secure Coding Modules. 2019 International Conference on Security and Management.

[6] Yuan, Xiaohong; Yang, Li; Jones, Bilan; Yu, Huiming; & Chu, Bei-Tseng. (2016) "Secure Software Engineering Education: Knowledge Area, Curriculum and Resources," Journal of Cybersecurity Education, Research and Practice: Vol. 2016: No. 1, Article 3.

[7] Long, F., Mohindra, D., Seaccrd, R. C., Sutherland, D. F., & Svoboda, D. (2012). CERT Oracle Secure Coding Standard for Java. Addison-Wesley.

[8] Long, F., Mohindra, D., Seaccrd, R. C., Sutherland, D. F., & Svoboda, D. (2014). Java Coding Guidelines. Addison-Wesley. Seacord, R. C. (2013). Secure Coding in C and C++. Addison-Wesley.

[9] Seacord, R. C. (2013). Secure Coding in C and C++. Addison-Wesley.

[10] Yu, H., Jones, N., Bullock, G., & Yuan, X. (2011). Teaching secure *software engineering: Writing secure code.* 2011 7th Central and Eastern European Software Engineering Conference (CEE-SECR), doi: 10.1109/CEE-SECR.2011.6188473.

[11] NSA CLARK Cybersecurity Library, https://clark.center/home.

[12] Lodgher, A and Yang J, 2017. Cyber Security Modules for Core, Major and Elective Courses in the BS CS Curriculum, NSA Grant, 09/17-08/18.

[13] A. Lodgher, J. Yang and U. Bulut, "An Innovative Modular Approach of Teaching Cyber Security across Computing Curricula," 2018 IEEE Frontiers in Education Conference (FIE), doi: 10.1109/FIE.2018.8659040.

[14] Lesh, R., & English, L. D. (2005). Trends in the evolution of models & modeling perspectives on mathematical learning and problem solving. ZDM: The International Journal on Mathematics Education, 37, 487–489.

[15] Lesh, R., & Doerr, H. M. (2003). Foundations of a models and modeling perspective on mathematics teaching, learning, and problem solving. In R. Lesh & H. M. Doerr (Eds.), Beyond constructivism: Models and modeling perspectives on mathematics problem solving, learning, and teaching (pp. 3-33). Mahwah, NJ: Lawrence Erlbaum Associates.

[16] Frank, B., & Kaupp, J. (2012). Evaluating integrative model eliciting activities in first year engineering. Proceedings of the 2012 Canadian Engineering Education Association Conference. Retrieved from http://www.academia.edu/2761700/Evaluating_Integrative_Model_Eliciting_Activities_in_First_Year_Engineering.

[17] William Newhouse, Stephanie Keith, Benjamin Scribner, Greg Witte, "National Initiative for Cybersecurity Education (NICE) Cybersecurity Workforce Framework," National Institute of Standards and Technology (NIST), 2017, https://doi.org/10.6028/NIST.SP.800-181.

[18] Lesh, R., Hoover, M., Hole, B., Kelly, A., & Post, T. (2000). Principles for developing thought-revealing activities for students and teachers. In A. Kelly & R. Lesh (Eds.), Research design in mathematics and science education (pp. 591-646). Mahwah, NJ: Lawrence Erlbaum and Associates.

[19] Hamilton, Lesh, Lester, Brilleslyper. (2008). Model-Eliciting Activities as bridge between engineering education research and mathematics education research, ASEE.

[20] Moore, T. J., Miller, R. L., Lesh, R. A., Stohlmann, M. S., & Kim, Y. R. (2013). Modeling in engineering: The role of representational fluency in students' conceptual understanding. Journal of Engineering Education, 102(1), 141-178.

[21] Kaupp, J., Frank, B., & Chen, A. (2014). Evaluating critical thinking and problem solving in large classes: Model eliciting activities for critical thinking development. Toronto, Canada: Higher Education Quality Council of Ontario.

[22] Lesh, R., & Yoon, C. (2004). Evolving communities of mind - in which development involves several interacting and simultaneously developing strands. Mathematical Thinking and Learning, 6 (2), pp. 205-226.

[23] J. Yang, B. Earwood, Y. Kim, & A. Lodgher. "Implementation of Security Modules with Model-Eliciting Activities in Computer Science Courses," 2020 ASEE Annual Conference Proceeding, DOI 10.18260/1-2—34776.

[24] Allan Collins, Diana Joseph, Katerine Bielaczyc, Design Research: Theoretical and Methodological Issues, The Journal of the Learning Science, 13(1), 15-42, 2004.

[25] Nuñez, A.-M. (2015). "Hispanic-Serving Institutions: Where are they now?" A commissioned paper presented at the meeting "HSIs in the 21st century: A convening" at the University of Texas El Paso.

[26] Security and Microsoft teams. Retrieved from https://docs.microsoft.com/en-us/microsoftteams/teams-security-guide.

[27] Moore, T. J., Guzey, S. S., Roehrig, G. H., Stohlmann, M., Park, M. S., Kim, Y. R., Callender, H. L., & Teo, H. J. (2015). Changes in faculty members' instructional beliefs while implementing model-eliciting activities. Journal of Engineering Education, 104(3), 279-302.

[28] Roehrig, G. H. & Luft, J. A. (2004). Inquiry teaching in high school chemistry classrooms: The role of knowledge and beliefs. Journal of Chemical Education, 81(10), 1510-1516.

[29] Corbin, J., & Strauss, A. (2008). Basics of qualitative research (3rd ed). Thousand Oaks, CA: Sage.

[30] Miles, M. B., & Huberman, A. M. (1994). Qualitative data analysis. Thousand Oaks, CA: Sage.

[31] Luft, J. A., Bang, E. J., & Roehrig, G. H. (2007). Supporting beginning science teachers. The Science Teacher, 74(5), 24-29.