Effective Learning of Cybersecurity Concepts with Model-Eliciting Activities

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

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
youngrae.kim@tamusa.edu

***Abstract*—As security is a crucial aspect in the process of developing software systems, software engineers must have a strong understanding of security concepts for an application being developed and tested. There has been a growing demand for these skills to be taught on all knowledge levels in computing courses. This paper builds on a study related to a series of security modules designed to meet that demand for teaching security concepts to students in computer science courses. Six small lessons in three security modules are implemented into a CS2 course, and the outcomes of this implementation are assessed. Each concept in the modules is broken up into a general description of the security problem, sample code written in Java, and sample code of the solution. Along with the security modules, an open-ended, problem-solving Model-Eliciting Activity (MEA) was developed as a project for students to demonstrate an understanding of the security concepts. Experimental studies were conducted to investigate the teaching effectiveness of implementing cyber security modules with the MEA project and students' experiences in conceptual modeling tasks in problem solving. After implementing the security modules with the MEA, students showed a good understanding of cyber security concepts, and the instructor's beliefs about teaching shifted from teacher-centered to student-centered. 41.7% of the developed solutions from the MEA groups showed a sufficient degree of creativity, and 58.3% of the solutions seemed suitable for real-world implementation. The initial activities leading to student developed solutions effectively prepared students within the scope of the course, but additional discussion and resources may be necessary to expand on creativity and practicality.**

Keywords—cybersecurity, cybersecurity education, computer science education, model-eliciting activity, MEA, cipher, Java, CS2, defensive programming

I. Introduction

Cyber security is a critical aspect in the design, development, and testing of software systems. As cyber-attacks continue to increase in frequency, complexity, and severity, it becomes more essential for software engineers to consider security at every point in a software's life cycle. Markettos et al. advocated that security must be considered from the ground up to build complex hardware and software systems constructed for the new course of vulnerabilities [1]. Saydjari emphasized the coupling of designing and implementing software systems alongside system risk analysis and management [2, 3]. For these reasons, we believe that students must learn early on how to develop software systems with security in mind and to continue building those skills at all knowledge levels in computer science [4, 5]. Many efforts have been made to provide secure coding guidelines [6, 7, 8, 9, 10]. However, in most colleges and universities, secure coding practices are still not treated as a core aspect of programming. Instead, secure coding practices are considered a specific concentration covered during students' junior and senior levels. Therefore, the focus of this experimental report is to emphasize the need for fundamental security concepts at the freshman level.

This paper reports the implementation and outcomes of incorporating cyber security modules with an MEA project in a freshman level CS2 course (Programming Fundamentals II). There are two experimental studies: 1) Effectiveness Study to investigate the teaching effectiveness of implementing cyber security modules with the MEA project, 2) Study of Problem Solving to investigate students' experiences in conceptual modeling tasks in problem solving. The studies are grounded in the adaptation of MEAs and Models and were Perspectives (MMPs) on learning and problem solving described in the following section. For the teaching effectiveness, emphasis is placed on the second semester (Fall 2020) with a review of the instructor's effectiveness in the first semester (Fall 2019) [24]. Discussion about both semesters is included to describe overall trends with instructor's attitudes towards student learning and teaching practices. For the study of problem solving, students' experiences in conceptual modeling tasks in problem solving and their understanding of course materials with the cyber security concepts were measured and interpreted. Student solutions from the Fall 2020 semester are assessed.

II. Cyber Security Modules and MEA

A. Cyber Security Modules

The cyber security modules for computer science courses were developed for a National Security Agency (NSA) grant project [11]. These modules are currently available at the NSA's CLARK Cybersecurity Library for public access [13]. The set of the first five modules were taught during the Fall 2019 semester in CS1 courses at two institutions, Texas A&M University-San Antonio (A&M-SA) and San Antonio College (SAC) as discussed in the prior research paper [24]. The other three modules with six lessons were taught in a CS2 course during the Fall 2020 semester at A&M-SA. These three modules were designed to introduce fundamental security concepts of defensive object-oriented programming in

TABLE I. INCORPORATION OF CYBER SECURITY MODULES

Chapter to Cover Module	Module#.Lesson(s)		NICE Category, SAs & KSAs	CWE
Ch. 6. A First Look at Classes	1.1 Declaring Fields		T0686, K0009 T0176, T0072 T0516, T0728 T0183	CWE-200
Ch. 8. A Second Look at Classes and Objects	1.3 Returning References	2 Validate Arguments		CWE-20
Ch. 10. Inheritance	1.2 Using Private Members			CWE-200
Ch. 11. Exceptions and Advanced File I/O	3.1 Checked Exceptions	3.2 Exceptions for Sensitive Information	K0005, A0092	CWE-434

beginner-level courses [11, 12]. They were also designed to be complete and independent to allow for easy integration into the course. Each module package consists of instructions, lab exercises with solutions, and assessment methods.

The modules deal with common object-oriented issues such as privacy, visibility, dependency of class members, valid use of method arguments among classes, and exception handlings. For example, students ensure that any changes made in a super-class must preserve all the program invariants that its sub-class depends on because failure to preserve dependences can cause security vulnerabilities. Data members of class that are exposed by declaring them as public or protected are prone to unexpected attacks. Vulnerability of such attacks can be reduced by increasing the privacy level to private declarations. Each lesson was presented in lectures as a general description of a security issue along with sample code of both the security issue and solution implemented in Java.

B. Model-Eliciting Activity (MEA)

MEAs are open-ended, problem-solving activities in which groups of three to four students work to solve realistic complex problems in a classroom setting [14]. For this study, the MEA was incorporated into the course as a semester project, which reflects six design principles: Reality, Model Construction, Model Documentation, Self-Assessment, Generalizability, and Effective Prototype [15]. In general, MEAs are designed based on the Models and Modeling Perspectives (MMPs) on learning and problem solving. MMPs draw on research related to constructivist views on learning from Piaget, Vygotsky, Charles S. Peirce, and John Dewey [16]. Student learning occurs in the development of models adhering to the six design principles. MEAs emphasize building solutions (models) for realistic problems situated in socio-cultural contexts that can also be applied to other problem situations and contexts [14]. In addition to building solutions for realistic problems, MEAs help improve metacognitive abilities through conducting multiple cycles of revisions [17]. Overall, MEAs provide effective teaching and learning in a student-centered context [8, 9, 18].

MEAs generally involve multiple iterations of expressing, building, testing and revising conceptual models [15]. This iterative process means developing tools for imaginary clients in realistic problems [14, 19]. This process consists of an individual activity and a group activity. In the individual activity, students are provided background information and work through a simplified problem by themselves. In the group activity, students use the knowledge and skills gained from the individual activity to prepare a solution (model) for a more complex problem in groups of three to four. MEAs also help make engineering students better problem solvers and provide experiential learning opportunities [17, 20, 21, 22]. MEAs are easily integrated into existing curricula [19]. This integration also made MEAs an ideal fit for seamlessly introducing cyber security concepts into the course content.

III. IMPLEMENTATIONGN OF SECURITY MODULES AND MEA

A. Incorporations of Cyber Security Modules

The concepts from the modules incorporated into lectures were taught in one section of CS2 courses in the Fall 2020 semester at Texas A&M University-San Antonio (A&M-SA). Then our experimental studies addressed the questions presented in section B C. 26 undergraduate students and one instructor at A&M-SA participated in this study. Table I outlines how the concepts of the modules were integrated with chapter materials, how they were related to CWE [26], and their Specialty Areas (SAs) and Knowledge, Skills, and Abilities (KSAs) in the NICE category [27]. The book used for the course was Starting Out with Java: From Control Structures through Objects, 7th Edition.

As shown in Table I, the Declaring Fields concept is covered in chapter 6 as classes were reviewed to prepare students for object-oriented programming. Part of this review included proper data hiding, which generally requires declaring fields as private and methods as public. The concepts of Returning References and Validate Arguments are both covered under the broader concept of aggregation (or association or composition) covered in chapter 8. The distinction between performing shallow copies or deep copies on objects is essential in understanding how to properly and securely handle objects as fields of classes. Deep copies must be performed when modifying an object field or when accessing an object field. Additionally, if a reference variable is one of the parameters of a method, a check must be performed to determine if that variable references a null object. Using Private Members emphasizes the need to declare inner classes as private as another means of adhering to proper data hiding.

This security concept is introduced in chapter 10 alongside other techniques for writing classes and constructing objects of classes, such as anonymous inner classes and lambda expressions. Understanding what Checked Exceptions are, including when try-catch blocks must be used in code, is a crucial first step in the materials covered in chapter 11. Students are also expected to understand what information must be provided to the user, including Exceptions for Sensitive Information.

B. Implementation of Model-Eliciting Activity (MEA)

1) Model-Eliciting Activity: To provide background information as well as a realistic context, students were given an article on the history of the Caesar cipher. Students also learned about the algorithms for encryption and decryption

TABLE II. INSTRUCTOR BELIEF OF TEACHING, LEARNING, AND ASSESSMENT

Interview		1st (beginning of fall 2019)	2nd (end of fall 2019)	3rd (beginning of fall 2020)	4th (end of fall 2020)
Teaching	Role as Instructor	(1)	(3)	(3)	(3)
	Maximize Student Learning	(2)	(2)	(2)	(3)
	What to Teach	(1)	(1)	(2)	(3)
Learning	How Students Learn Best	(2)	(3)	(2)	(2)
	Learning Occurs	(3)	(3)	(5)	(5)
Assessment	When Students Understand	(3)	(5)	(2)	(4)
	When to Move on	(1)	(2)	(4)	(4)

TABLE III. INSTRUCTOR CHANGE OF BELIEFS OVER THE TWO YEARS

Interviews	Traditional (1)	Instructive (2)	Transitional (3)	Emerging (4)	Constructivist (5)
1st Interview	***	**	**		
2nd Interview	*	**	***		*
3rd Interview		****	*	*	*
4th Interview		*	***	**	*

for the Caesar cipher, affine cipher, and block cipher. Students then were asked to implement these encryption and decryption algorithms in Java. Solutions were provided and discussed after students completed the individual activity.

2) Group Activity: Student groups were given the task of designing and implementing a unique cipher algorithm based on the principles learned from the individual activity [16, 24]. Each group is composed of two or three students. Students were given background information about the need to design and implement a entirely new encryption algorithm to help protect against a man-in-the-middle attack. As deliverables, student groups were required to prepare 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. They had to implement an encryption algorithm in Java based on the designed algorithm, and had to present their solutions to the rest of the class. Students prepared recordings of their presentations that were submitted through the course Blackboard.

C. *Effectiveness Study and Problem Solving*

The inclusion of the MEA sets the foundation for two major experimental studies: 1) Effectiveness study to investigate the teaching effectiveness of implementing cyber security modules and the MEA, and 2) Study of Problem Solving to investigate students' experiences in conceptual modeling tasks in problem solving. The design experiment methodology is used to study the effectiveness of incorporating cyber security modules and the MEA [23]. The effectiveness study is guided by the following question.

1. 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?

The study of problem solving is guided by the following questions:

1. Whether students conceptually connected with the project along with the course contents, if not, what misconceptions did the students have?
2. Are the solutions and ideas applicable to the implementation for real-world applications?
3. Are developed solutions creative?

IV. RESULTS AND DISCUSSION

To address the questions for the effectiveness study, quantitative and qualitative data were collected through semi-structured pre- (beginning of the semester) and post-interviews (ending of the semester) of the instructors. The interview questions for the instructors were modified and adapted from previous studies [1, 25]. The following questions were included in the pre-interviews:

- How do you describe your role as the instructor?
- How do your students best learn engineering?
- How do you maximize student learning in your classroom?
- How do you know when your students understand?
- How do you decide what to teach or what not to teach?
- How do you decide when to move on to a new topic in your class?
- How do you know when learning is occurring in your classroom?

The following questions were included in the post-interviews:

- What are some changes in your classrooms after the use of MEAs for cyber security modules?
- What are some differences between your expectation and your observation in the student work through the use of MEAs for cyber security modules?

TABLE IV. CIPHER ALGORITHM SOLUTIONS

Group	Solution	Caesar Cipher, Affine Cipher, Block Cipher, or/and Others
1	Vigenere Cipher	Vigenere Cipher
2	RSA cryptography	RSA cryptography
3	Caesar Cipher	Caesar Cipher: B = (A – 3) mod 26
4	Base 64 conversions	Base 64 conversions
5	Caesar Cipher	Caesar Cipher: B = (A – 3) mod 26
6	Affine Cipher	Affine Cipher: B = (3A – 5) mod 26
7	Vigenere Cipher	Vigenere Cipher
8	Caesar Cipher	Caesar Cipher: B = (A – 3) mod 26
9	Caesar Cipher	Caesar Cipher: B = (A – (U + R)) mod 26, U = user input, R = randomly generated number
10	Caesar Cipher and Block Cipher	Caesar Cipher: B = (A – 2) mod 26, Block Cipher: size 4
11	Caesar Cipher	Caesar Cipher: B = (A – 8) mod 26
12	Block Cipher and XOR	Block Cipher: size 5

To address the questions for the study of problem solving, the outcomes of the incorporation of the cyber security modules and MEA were collected and observed from the student groups at the end of the semester. The outcomes of the MEA included both the source code and presentations.

A. Results of Effectiveness Study

Table II summarizes the results for the instructor interview questions before and after the Fall 2019 and 2020 semesters. The instructor indicated that he was an instructor who displayed a combination of "traditional" and "instructive" traits at the beginning of the Fall 2019 semester. The instructor viewed his role as a teacher "to cover general concepts" (traditional). To maximize student learning, he provided students with "real-world examples or demonstrations" (instructive). He believed that students best learn to engineer when they are given opportunities for "practicing a lot" (instructive). The instructor also tried to measure student understanding through "holding a conversation about the topics being discussed" with students during the lecture (transitional).

After the first implementation of the cyber security modules with the MEA project, the instructor described his job as an instructor as "[giving] [students] information outside of the exams and labs to help them understand the materials" (transitional). He believed that he could maximize student learning by "constantly asking questions" and "having [students] step through a program" (instructive). He felt that students learn best "from feedback" and "from smaller, more controlled hands-on projects" (transitional). In order to assess student understanding, he considered if his students could "reciprocate and ask questions beyond course materials" (experienced constructivist).

At the beginning of the Fall 2020 semester, the instructor indicated that he was an instructor having "instructive", "transitional", "emerging constructivist", and "experienced constructivist" views. The instructor viewed his role as a teacher to guide students in developing "technical knowledge" and "critical thinking" skills (transitional). To maximize student learning, he provided students with "theoretical and real-world examples" and asked students questions to "see if they are aware of how to take the explanation" (instructive). He believed that students best learn to engineer when they are given opportunities through "practical examples" (instructive). He knew when learning was occurring in his students "whenever the students can demonstrate to [him] that they actually understand the material" and "whenever the students will ask [him] questions that go a little bit beyond the scope of what [he] covered" (experienced constructivist).

After the second implementation of the cyber security modules with the MEA project, the instructor described his job as an instructor to guide students in developing "the basics, the fundamentals [of knowledge] first, and then start getting them to apply that critical thinking later" by giving them "not just material in the course but maybe outside resources, kind of anything that exists in the field at the time" (transitional). He believed that he could maximize student learning by "essentially providing [students] with the resources" and "to practice the concepts" and "giving them the opportunity to ask questions and be able to work on that" (transitional). "Whenever the students feel pretty confident about what it is that they have to do" and "they start asking questions that go a little bit beyond the scope of what we've been covering up to that point," the instructor knew whether learning was occurring in his classes (experienced constructivist). Table III shows the general shift of responses from "traditional" towards "constructivist." This indicates a shift in the instructor's attitudes towards a more student-centered view about teaching, learning, and assessment because of the cyber security modules with the MEA project.

B. Results of Problem Solving Study

To assess the results of the study of problem solving, the solutions to the MEA project were assessed. The MEA project served as a practical application of most of the concepts, either security modules or standard course content, presented in the lecture materials. Reasonable solutions to the MEA project indicated a strong understanding of the security concepts in the previous security modules.

1) Overview of MEA solutions: For each of the groups involved in the MEA, there were four outcomes: 1) using one of the algorithms learned in the individual activity; 2) combining two of the algorithms learned in the individual activity; 3) combining one of the algorithms learned in the individual activity with an entirely new algorithm; or 4) using an entirely new algorithm. This order highlights the least to most inventive approaches to solving the problem, and slightly highlights the least to most practical ideas. The most

TABLE V. ANALYSIS RESULTS OF PROBLEM SOLVING QUESTIONS

Gr.	Solution	Q1	Q2	Q3
1	Vigenere Cipher	Good understanding of expanding the initial concept of Caesar Cipher	**Yes**	**Yes**
2	RSA cryptography	Sufficiently well-researched solution Requires the use of multiple classes from the Java API	**Yes**	**Yes**
3	Caesar Cipher	Uses try-with-resources blocks to handle reading and writing text files Implemented separate custom classes for different components	No	No
4	Base 64 conversions	Interesting use of Base64 class but potentially lacks an understanding of securely encrypting text Implemented GUI based on JavaFX concepts	No	**Yes**
5	Caesar Cipher	Noted difficulties with using character arrays and performing the math related to encryption and decryption shifts	No	No
6	Affine Cipher	Incorrectly uses try-catch blocks on code that does not throw checked exceptions Possibly misunderstanding about the terms encryption and decryption	No	No
7	Vigenere Cipher	Good understanding of expanding the initial concept of Caesar Cipher (similar to group 1)	**Yes**	**Yes**
8	Caesar Cipher	Researched concepts covered in the book, including use of RandomAccessFile class Considered principles beyond scope of the class, such as user flexibility and algorithm difficulty	No	No
9	Caesar Cipher	Randomly generated numbers also demonstrate a good understanding of Caesar Cipher, but are not necessarily a more practical solution	No	**Yes**
10	Caesar Cipher & Block Cipher	Good understanding of how to implement multiple algorithms to improve security	**Yes**	**Yes**
11	Caesar Cipher	Uses throws clauses rather than try-catch blocks Implemented separate custom classes for different components (similar to group 3)	No	No
12	Block Cipher & XOR	Effectively uses a swapping algorithm to rearrange blocks of text Good use of XOR operations, which are not directly introduced in the course	**Yes**	**Yes**
		Percentage of Yes	**41.7 % (5/12)**	**58.3% (7/12)**

inventive ideas presented generally made for the most practical solutions in terms of security.

Table IV shows the algorithms involved in each group's proposed solution along with additional clarifications as needed about that specific implementation. Overall, out of twelve groups, five groups utilized only the Caesar Cipher (1 modified and 3 direct use), one group utilized only the Affine Cipher (1 direct use), no groups utilized only the Block Cipher, and seven groups presented either combinations or entirely new algorithms. Of these remaining groups, one group utilized a combination of the Caesar Cipher and Block Cipher, one group utilized a combination of the Block Cipher and performing XOR operations on each character, two groups utilized the Vigenere Cipher, one group utilized base 64 conversions, and one group utilized RSA encryption.

2) MEA Assessment: Column Q1 in Table V shows the analysis results of investigating the first question in section III C: *Whether students conceptually connected with the project along with course contents, if not, what misconceptions did the students have*? Overall, the students demonstrated a strong understanding of the concepts covered in the lecture materials of the course, including the security modules. Solutions included the use of multiple classes (chapters 6 and 8), the use of try and, more specifically, try-with-resources blocks (chapter 11), the use of the RandomFileAccess class (chapter 11), and the construction of a GUI with JavaFX (chapters 12 and 13).

Algorithm implementations were adequately derived from the individual activity and showed a good understanding of character arrays and loops, character conversions, and arithmetic operations. Conversely, students noted difficulties with deciding on an appropriate algorithm and, in some cases, learning how to properly perform shifts for the encryption and decryption process. Other issues identified in the final solutions included the use of throws clauses rather than try-catch blocks and the use of different algorithms for encryption and decryption. In the latter case, the solution included both the encryption and decryption for the Caesar Cipher as "encryption" and both the encryption and decryption for the Affine Cipher as "decryption." This suggests the student understood how to implement the algorithms but had a weak understanding of the terms, encryption and decryption in theory.

Column Q2 in Table V shows the analysis results of investigating the second question: *Are the solution and ideas applicable to the implementation for real-world applications?* The criterion for this question was an adequate level of security. Consideration was given to the level of programming for the course, so an adequate level of security was considered any algorithm that cannot be broken using frequency analysis or an equally effective technique. Some of the solutions (41.7% or 5/12) were deemed suitable for real-world implementation. These solutions included the two Vigenere Ciphers, the RSA encryption, and the two combinations using the Block Cipher.

While frequency analysis is one of the techniques used to break some of these algorithms, such as Vigenere, the necessity of other more complex techniques was sufficient to consider these algorithms reasonably secure. Of the seven solutions deemed not suitable for real-world implementation, most (6/7) were primarily variations on the Caesar and Affine Ciphers, which do not provide a sufficient level of security. The remaining solution that was deemed not suitable was the

base 64 conversions, which is simply broken by reverting these conversions. This latter set of solutions demonstrates a misconception about what constitutes secure data because the students have not been exposed to techniques for breaking these encryption algorithms, such as the previously noted frequency analysis.

Column Q3 in Table V shows the results of investigating the third question: *Are developed solutions creative*? Solutions are considered creative if there is a novel transformation from the ciphers presented in the individual activity. Novel transformations include modifications, combinations, or entirely new implementations. While they were at different levels of creativity, the majority of the ideas (58.3% or 7/12) were creative. The five solutions that were not considered creative were direct implementations of the Caesar and Affine Ciphers. The remaining seven solutions included a modification of the Caesar Cipher using randomly generated numbers, the two Vigenere Ciphers, the RSA encryption, the base 64 encryptions, and the two combinations using the Block Cipher.

V. Elaboration on Creative Solutions

To further elaborate on the creativity of these solutions, we consider the two most noteworthy solutions: the RSA encryption by Group 2 and the Block Cipher with XOR operations by Group 12 on each character. Students in Group 2 researched various encryption algorithms to arrive at the solution of RSA encryption. To properly understand the implementation of this solution, these students also had to study the various classes from the Java API related to performing RSA encryption, including KeyPair, KeyPairGenerator, PublicKey, and Signature. The StandardCharsets class was also used to properly handle character conversions for encrypted text. Students also provided a theoretical explanation of RSA encryption, which indicated a good understanding of the principles behind this encryption standard. Overall, this solution demonstrates a significant degree of novelty compared to the Cipher algorithms presented in the individual activity.

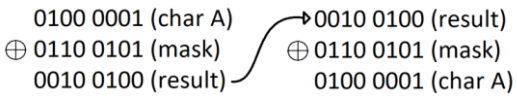

Fig. 1. Perforimg XOR on a target set of bits

Students in Group 12 utilized a combination of the Block Cipher presented in the individual activity as well as XOR operations performed on the individual characters. The latter characteristic of this solution is the primary reason this is considered notably creative. The XOR operation is highly useful in encryption due to two important factors: 1) the same key is used for both encryption and decryption; and 2) this operation is highly resistant to brute force attacks. Fig. 1 demonstrates performing XOR on a target set of bits using the same bit mask twice. The result is restoring the original set of bits, demonstrating the usage of the same key for encryption and decryption. The combination of a Block Cipher and XOR operations is similar to steps involved in the SHA-2 algorithm.

VI. Conclusion and Future Work

This paper has presented the outcomes of implementing three cyber security modules with the MEA project in a freshman level CS2 course at A&M-SA during the Fall 2020 semester. The outcomes are from the two experimental studies: 1) Effectiveness Study to investigate the teaching effectiveness of implementing the cyber security modules and MEA, and 2) Study of Problem Solving to investigate students' experiences in conceptual modeling tasks in problem solving.

For the effectiveness study, the instructor's attitudes towards student learning were assessed over both semesters. For the second study, the results of student outcomes through the MEA from the student groups were observed in the Fall 2020 semester. Students' conceptual modeling tasks in problem-solving were assessed. Instructor effectiveness was assessed based on the change in attitudes towards student learning and their teaching practices. Results from the Fall 2019 and Fall 2020 semesters indicated a shift from teacher-centered views to student-centered views from the instructor. The answers to the interview questions reflect a positive impact on their beliefs and decisions about teaching, learning, and assessment.

For the study of problem solving, all students showed a good understanding of cyber security concepts related to the MEA as the semester project. Most student groups used modified an existing Cipher, used a combination of Ciphers, or used an entirely new algorithm. 58.3% (7/12) of the developed solutions from the student groups with the MEA project show a sufficient degree of creativity. 41.7% (5/12) of the solutions do seem suitable for real-world implementation.

To further explore ways to increase creative and practical solutions, additional criteria will be provided in the group activity. Students will be exposed to frequency analysis and must consider solutions that work around this added difficulty. In order to investigate the overall findings further, the sets of remaining security modules with other MEAs will be designed and incorporated into later courses, such as Data Structures and Algorithms, Database Systems and Software Engineering.

Acknowledgment

This work was partially supported by the National Science Foundation (NSF)'s grant project entitled "Recruiting and Retaining Students into Computing" under the award #1832433. This material is based upon work supported by the Grant. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

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), San Jose, CA, USA, doi: 10.1109/FIE.2018.8658433.

[5] Lodgher, A. , Yang, J. and Bulut, U., "An Innovative Modular Approach of Teaching Cyber Security across Computing Curricula,"

2018 IEEE Frontiers in Education Conference (FIE), doi: 10.1109/FIE.2018.8659040.

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

[7] 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.

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

[9] `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.

[10] Yang, J., Lee, Y., Hernandez, A., & Sanchez, J. "Evaluating and Securing Text-Based Java Code through Static Code Analysis," Journal of Cybersecurity Education, Research and Practice, Vol. 2020: No. 1, Article 3.

[11] Lodgher, A and Yang J, 2017, Cyber Security Modules for Core, Major and Elective Courses in the Bachelor of Science (BS) Computer Science Curriculum, NSA Grant, Sept 2017-Aug 2018.

[12] Yang, J. and Lodgher, A. "Fundamental Defensive Programming Practices with Secure Coding Modules," 2019 International Conference on Security and Management, ISBN: 1-60132-509-6.

[13] NSA CLARK Cybersecurity Library, Retrived from https://clark.center/home.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[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. Retrieved from http://www.heqco.ca/SiteCollection Documents/Formatted%20Queen%27s_Frank.pdf.

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

[24] Yang, J., Earwood, B., Kim, Y., and Lodgher, A., "Implementation of Security Modules with Model-Eliciting Activities in Computer Science Courses," 2020 ASEE (American Society for Engineering Education) Annual Conference Proceeding, DOI 10.18260/1-2—34776.

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

[26] 2021 CWE Top 25 Most Dangerous Software Weakness, https://cwe.mitre.org/top25/archive/2021/2021_cwe_top25.html.

[27] NIST Special Publication 800-181, National Initiative for Cybersecurity Education (NICE) Cybersecurity Workforce Framework, https://doi.org/10.6028/NIST.SP.800-181r1.