Electronic Voting Technology Inspired Interactive Teaching and Learning Pedagogy and Curriculum Development for Cybersecurity Education

Ryan Hosler, Xukai Zou, Matt Bishop

Electronic Voting Technology Inspired Interactive Teaching and Learning Pedagogy and Curriculum Development for Cybersecurity Education

Ryan Hosler[1][0000−0002−2169−5126], Xukai Zou[1][0000−0001−5762−8876], and Matt Bishop[2][0000−0002−7301−7060]

[1] Indiana University Purdue University Indianapolis, 420 University Blvd, Indianapolis, IN 46202, USA rjhosler@iupui.edu xkzou@cs.iupui.edu
[2] University of California Davis, 1 Shields Ave, Davis, CA 95616, USA mabishop@ucdavis.edu

Abstract. Cybersecurity is becoming increasingly important to individuals and society alike. However, due to its theoretical and practical complexity, keeping students interested in the foundations of cybersecurity is a challenge. One way to excite such interest is to tie it to current events, for example elections. Elections are important to both individuals and society, and typically dominate much of the news before and during the election. We are developing a curriculum based on elections and, in particular, an electronic voting protocol. Basing the curriculum on an electronic voting framework allows one to teach critical cybersecurity concepts such as authentication, privacy, secrecy, access control, encryption, and the role of non-technical factors such as policies and laws in cybersecurity, which must include societal and human factors. Student-centered interactions and projects allow them to apply the concepts, thereby reinforcing their learning.

Keywords: electronic voting · interactive teaching and learning · curricula.

1 Introduction

Cybersecurity defends against attacks that plague individuals and organizations daily; hence, cybersecurity has become integral to society. Cyberattacks and defenses are based on a combination of theoretical and practical knowledge and understanding, which often challenges students studying the foundations of cybersecurity. One way to excite interest is to tie the theory and practice to important events such as elections. Safe and secure elections are imperative for a democracy to function. The uniqueness and ubiquity of elections and the widespread use of E-voting systems emphasize the special role that E-voting technology can play in academic cybersecurity education in both college and high school.

Voting has a unique combination of security and integrity requirements [1]. For example, in secret-ballot voting, the most widely used voting scheme (and one used exclusively in the United States), a key security requirement is that the voter cannot be associated with a particular ballot, *even if the voter wishes to disclose that relationship* — very different than the security requirements for a banking ATM, for example, where

a customer must be able to prove their association with a transaction. Thus, E-voting technology involves many specific and sometimes conflicting requirements. The topic covers a large knowledge base of cryptography, system security, and network security. Studying a cryptographic-based network E-voting system will cover many aspects of computer security and information assurance in a way that students will see both the theoretical and practical benefits and disadvantages of various techniques.

From a pedagogic point of view, interactive teaching and learning methodologies have become more and more attractive nowadays. Interactive learning's impact on student learning outcomes, particularly in cybersecurity education, has been proven effective in both theoretical research [2, 3] and practical systems such as Clicker. Moreover, a case study showed that E-voting technology can be used to achieve student learning objectives satisfying ABET (Accreditation Board for Engineering and Technology) requirements [1], as does research [4–6].

In this paper, we propose an E-voting based student-centered interactive teaching and learning framework and curriculum for cybersecurity education based on a recent E-voting technique in [7]. Section 2 discusses other work involving cybersecurity education and electronic voting. Section 3 explains the modules used in the mutually restraining E-voting system and a mapping to the Cybersecurity Curricula 2017 (CSEC 2017) [8] defined 8 knowledge areas (KAs). Section 4 describes the interactive components of the modules. Then section 5 provides concluding remarks and potential for future research.

2 Related work

Interactive learning's impact on students, particularly in cybersecurity education, has been proven effective in both theoretical research [2, 3] and practical systems such as Clicker. Education tools and programs include general security [9–11], enhancing security education using games [12–14], and strengthening security education by exploring some specific aspect of security such as web browsers [9], software security [15–18], IoT security [19], cyber-physical system security [20,21], and network security [22,23]. Here, the specific aspects are those of elections and E-voting systems.

Elections have many stringent requirements such as completeness, correctness, security, confidentiality, auditability, accountability, transparency, simplicity, usability, accessibility, and fairness. Moreover, some cybersecurity topics appear to conflict with each other, such as anonymity and verifiability. Using E-voting technology to teach computer security enhances student learning outcomes [4–6]. Typical examples of such efforts are individual E-voting courses [24, 25].

Bishop and Frincke [1] identify five aspects of E-voting useful for a computer security course: (1) identifying security-relevant requirements; (2) understanding specification; (3) understanding confidentiality, privacy, and information flow; (4) understanding human elements; and (5) establishing confidence in the final tallies. These lead to 11 learning outcomes required by the Accreditation Board for Engineering and Technology (ABET) [1, p. 54]. For example, ABET outcome A, "[a]n ability to apply knowledge of mathematics, science, and engineering" occurs from material throughout the E-voting lessons, especially in the sections on establishing confidentiality and understanding the

human element, and outcome K "[a]n ability to use the techniques, skills, and modern engineering tools necessary for engineering practice" comes from the sections on requirements, specification, and confidence. The E-voting system used for the proposed curriculum is detailed in the next subsection.

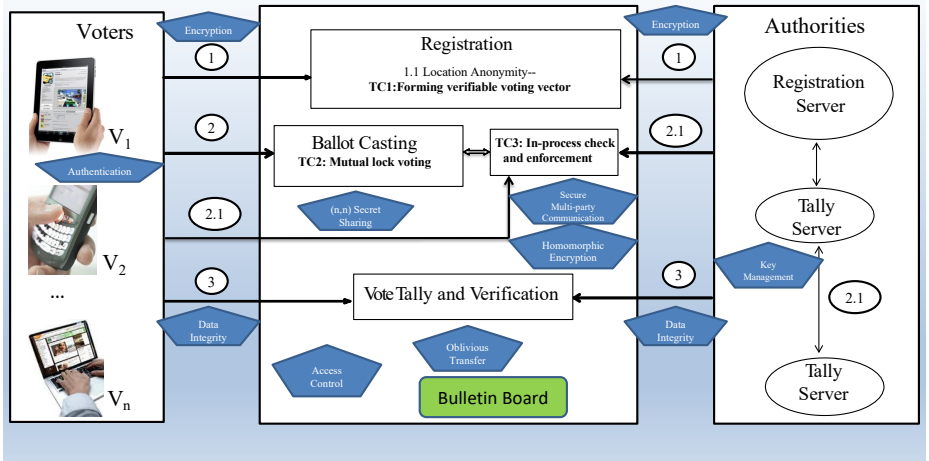

Fig. 1: Mutual restraining E-voting architecture (and mapped topics in pentagons).

2.1 Mutual-Restraining E-voting

The *mutual restraining electronic voting and election protocol* [7, 26] balances multiple parties with conflicting interests based on a few simple cryptographic primitives and assumptions. The protocol consists of three technical components (TC): a universal verifiable voting vector (TC1), forward and backward mutual lock voting (TC2), and in-process checking and enforcement (TC3). Figure 1 shows this architecture. As its underlying concepts and mechanisms are fundamentally different from other E-voting technologies, the mutual restraining E-voting technique is ideal for an interactive pedagogical framework.

Consider an election with N voters and M candidates for an office; each voter can vote for exactly 1 candidate. One can view the votes as being unique instances of an $L = N \times M$ matrix, with each voter being associated with a unique row. The mutually restraining E-voting protocol uses this idea as the basis for ballots. There are $N + 2$ entities, the voters and two *collectors* (e.g., the two tally servers in Figure 1). The mutual restraining E-voting protocol proceeds as follows:

1. The voter registers using a location anonymity scheme (LAS). The LAS gives the voter a unique row number known only to the voter. The voter Vi then sets to 1 the element corresponding to the candidate they wish to vote for (let the position be ${L_c}^i$) and all other elements of the row (as well as all elements of all other rows) to 0 (Step 1 of Figure 1). Moreover, the LAS is robust to a malicious participant deliberately inducing collisions by choosing a location that is already occupied by another voter [7].

2. The voter casts the ballot by generating two numbers $v_i = 2^{L-{L_c}^i}$ and $v'_i = 2^{{L_c}^i-1}$ from their $N \times M$ matrix. One collector generates $\frac{N}{2} - 1$ shares and the other, $\frac{N}{2}$ shares. They use an (N, N) secret sharing scheme to give each other voter a share, and send the sum of those shares to the voter. From this, the voter generates two secret ballots p_i and p'_i, the sum of shares held by V_i's secret ballot, from v_i and v'_i respectively, and the information received from the collectors (Step 2 of Figure 1). Ballots p_i and p'_i are made public; v and v' are secret and only known to the corresponding voter.
3. The two collectors now compute $P = \sum_{i=1}^{N} p_i$ and $P' = \sum_{i=1}^{N} p'_i$. When converted to binary vectors, the two vectors will be the complement of each other, if everyone was honest and no errors were made (Step 3 of Figure 1).
4. Each voter's ballot is checked for validity. When the voter sends the outputs of functions of v_i and v'_i to the collectors, the nature of these functions precludes their inversion but allows the collectors to check that both numbers are valid and hence the voter has cast exactly 1 vote. This validity is enforced since $v_i \times v'_i = 2^{L-1}$ regardless of which candidate voter V_i voted for. Hence, any deviation from correctly casting a ballot will be detected. When the secret ballots are published, they also can be validated using the outputs of the functions (Step 2.1 of Figure 1).
5. A web-based bulletin board system posts the aggregate votes as they are computed. It also shows the aggregation of the secret ballots once all are posted.

Any voter can verify their vote was counted correctly by examining the ballots and the aggregate vote by examining their sum, just as the collectors do.

This scheme and its protocols use cryptographic concepts throughout. It also relies on system and human security to ensure the integrity of the votes and of the inability to associate a voter with a ballot.

As an example, consider Step 2, above. If access to the server can be blocked, or the shares of the secrets associated with the voters computed, then the election will fail because ballots will show up as corrupted. More tellingly, if the web server bulletin board can be corrupted, the voters may believe the election was not run correctly or the final tallies are wrong, when in fact they were not. In an election, this lack of credibility corrupts the result as thoroughly as if the votes were actually changed.

Non-technical factors also come into play. In the US, many states disallow voting systems to be connected to a public network (or any network) while voting is underway. If voters use a smart phone or home system, an attacker can take advantage of their vulnerabilities to alter the voter's vote. Finally, management issues abound, particularly when one realizes poll workers are often people with limited to no experience with computing.

3 Modules mapped directly from E-voting

Educational modules can be tied to the development and use of the above-mentioned E-voting systems. The following ten examples modules form the core of a course in computer security. Each can be mapped to the mutually restraining E-voting system, as Figure 1 illustrates. In that Figure, pentagons indicate how each of the cybersecurity

topics relates to different activities and components of the mutual restraining E-voting system.

Module 0: E-voting and cybersecurity topic mapping. This introductory module covers the election process, requirements derivation and validation, and an examination of what security mechanisms are required to protect the system and voters. This includes cryptographic mechanisms and their use in protecting the integrity of data and transmissions. Via voting as a real world application, the instructor can lead students to discuss its properties and security requirements. Mapping different components of the system leads to the cryptographic primitives and security concepts of the system.

Module 1: User and system authentication. This module discusses the initialization of authentication information, a topic often overlooked but critical to the correct functioning of systems. Data poisoning can result in compromised systems, in this case compromised election results. As an example, voters must register, prove both their identity and place of residence in order to be able to vote; in some places, they must prove identity when they are given a ballot. If the former is incorrect, then the legitimate voter cannot prove that they are registered to vote, and hence are disenfranchised. This E-voting system allows remote voting, so transmitting trusted authentication information over untrusted channels must also be considered. Zero-knowledge proofs and other, more widely used, methods of authentication [27–30] are covered here.

Module 2: Confidentiality. Confidentiality protects the interaction of the voter with the ballot until it is cast. This includes the exchange of information to validate the cast ballot as legitimate without exposing any information about the voter beyond their being authorized to vote and that they have submitted one ballot. Therefore this module covers cryptography for secrecy, including secret key and public key cryptosystems. In addition, when the voter votes, malware in the E-voting system could transmit the ballot to a third party. With respect to cryptography, an intruder could corrupt the negotiation of the cryptographic protocol to be used or corrupt the cryptographic keys, the former enabling eavesdropping and the latter a denial of service or a masquerading attack. Thus, system security controls must supplement the cryptographic mechanisms.

Module 3: Data integrity and message (sender) authentication. Cryptographic methods such as one-way functions and digital signatures can be used to protect cast ballots and transmissions from being tampered with. System access controls augment these by protecting the integrity of the systems and software involved, as well as the data (ballots) stored there, and the systems that receive and process the votes. Procedural controls protect the integrity of the overall election and the results of the election. The latter can be analyzed semiformally [31]. These techniques are also used to ensure that the correct certified software is loaded onto the E-voting systems, and that once loaded the software is not altered or tampered with.

Module 4: Key management. Central to the use of all cryptosystems is cryptographic key generation and management. This topic covers secret key management, public key management, and group key management. Underlying this is the generation of truly random numbers, which typically requires unpredictable data to be gathered from various sources such as system hardware. It also requires that the systems on which the keys are generated and stored be tamperproof, as otherwise an adversary can substitute their

own keys, or corrupt the key generation program to ensure the keys are reproducible. These properties hold in many environments, not just voting.

Module 5: Privacy and anonymity. How a voter voted must be known only by that voter, and they cannot be able to prove how they voted to anyone. In addition to privacy and anonymity principles and mechanisms such as Mixnets [32–36], this topic includes repudiation (leading to non-repudiation). Legal considerations also drive mechanisms. For example, in the US, some states forbid any unique markings on ballots until the ballot is cast, for reasons of privacy; this inhibits the use of some protective mechanisms. Also, if a Mixnet goes outside one jurisdiction, another jurisdiction (nation) can block the transmission of those votes, so that must be balanced with untraceability requirements.

Module 6: Access control. E-voting systems have access control policies and mechanisms to regulate access for all entities including voters, authorities, and third parties. These range from the technical, such as the use of role-based access control, to the procedural, such as who can view cast ballots after the election and how long those ballots must be preserved. Conflicts of interest also affect these policies. As noted above, procedure analysis can semiformally analyze the procedures used to enforce those requirements. All these techniques are covered.

Module 7: Secure group/multi-party communication and secret sharing. The mutual restraining E-voting technique uses (n, n) secret sharing for n voters to exchange votes' shares and to obtain the sum of votes. Thus, this topic covers secret sharing and secure multi-party communication schemes. Protection of the shares and their transmissions are also relevant here, as are commitment schemes that allow one to commit to a chosen value while keeping it hidden from others [37–40].

Module 8: Secure multi-party computation and homomorphic encryption. Secure multi-party computation (particularly multiplication) is used among authorities to prevent any voter from casting multiple votes. Secure two party multiplication is implemented via homomorphic encryption. Threshold cryptography prevents authorities from colluding and guarantees that a certain number of authorities can perform necessary tasks. [41–44] These advanced types of cryptography are covered in this topic.

Module 9: Attacks and defenses. Attackers will attack all parts of an E-voting system including the voting and vote-tallying systems, communication channels, and the vote reporting mechanisms. Threat modeling, and the derivation of requirements from them, enable developers and election officials to anticipate these attacks and create appropriate defenses as well as detection methods for when those defenses fail. This leads to an analysis of potential attacks and how to detect and handle intrusions.

Modules 1,2, and 3 cover basic cybersecurity topics; modules 4, 5, and 6, intermediate cybersecurity topics; and modules 7, 8, and 9 cover advanced cybersecurity topics. The instructor can adjust the level of detail, and specific selection of cybersecurity topics, as they feel appropriate for their class.

3.1 Relationship to CSEC2017

The Cybersecurity Curricula 2017 (CSEC 2017) [8] defines 8 knowledge areas (KAs), each of which consists of knowledge units and essential learning outcomes. The modules present an avenue for teaching many of those learning essentials.

As an example, elections are governed by both laws and regulations. These vary among jurisdictions, so the instructor can begin by reviewing the local laws and how those constrain the design. In the US, the laws in all jurisdictions require that no voter can be associated with the cast ballot, which means that no one, including the voter, can say which cast ballot is that voter's (Societal and Organizational learning essentials). How this is done varies, and the students can brainstorm about different ways to protect the privacy and secrecy of the ballots (Human and Societal learning essentials). How these rules affect remote voting, and the construction of the systems, are other interesting areas to discuss; in some cases, certain cryptographic mechanisms would violate laws.[3]. Then the instructor can segue into translating these requirements into software constraints (Software learning essentials) and hardware constraints (Component, Connection, and System learning essentials).

As another example, in the context of elections and E-voting, user authentication covers all of the essential learning objectives of the Data KA; it also includes the Human KA identity management learning essential and the authentication part of the System KA. The types of proof of identity needed at the polling stations vary from a voter ID card to a simple verbal statement and recognition by a poll worker, and so relate to the cyberlaw learning essentials of the Societal KA. In all jurisdictions, impersonating a registered voter is a crime, leading to cyberlaw considerations (in the Societal KA).

4 Module format and Construction

Interactive learning involves students' active participation and engagement. Four interactive learning modules, along with their formats and constructions, are described below. The instructor can use these to cover individual topics or a mixture of topics above.

4.1 Interactive lecturing

Interactive activity engages the instructor and students in a controlled dialogue. The instructor, acting as the tallying authority, poses questions during lecturing on the bulletin board. Students, acting as voters, respond their answers via the E-voting system, and the system tallies and shows the results. The tallied answers can guide the instructor so they can tailor their lecture to the learning needs of the students.

This module can apply to different topics. With the implemented interface modules, the instructor sets questions, the students respond to questions, and the system computes and displays the students' responses. Moreover, the instructor can ask True/False, Yes/No, or multiple choice questions. For challenging questions, instructors can present essay prompts requiring students to provide justification and analysis.

In summary, this interactive lecturing module provides user friendly and flexible options for an instructor to engage students and gauge how well the students are learning. The knowledge gathered by assessing student responses will let the instructor know whether to advance to the next topic or stay on the present topic.

[3] For example, the California Election Code states "it is [to be] impossible to distinguish any one of the ballots from the other ballots of the same sort." [45, §13202]

4.2 Interactive class projects

This set of activities have the students interact with one another and with the system, both in and out of class. Interactive class projects are typically cybersecurity topic-related and in many cases involve multiple topics.

Class project design principles. The overall purpose of interactive class projects is to gain first-hand experience with security concepts and principles. The projects described below have students implement and analyze different components of the E-voting system and, ideally, integrate these components to form a complete E-voting system. Several approaches can be used to design class projects.

- A completely runnable mutual restraining E-voting system is provided at the beginning of the class for use by students during the class. This allows students to actively engage in the material from the beginning. During the course of the class, students will be asked to implement parts of the system to replace the corresponding existing ones in a cut-and-paste manner.
- Attacker and defender class projects have students act as attackers who find and exploit vulnerabilities and as defenders who try to thwart them. This includes both implementation and operation vulnerabilities.
- Class projects such as implementing vote casting, vote tally, and verification allow students to study trade-offs. They also can check that the system with the newly-implemented modules complies with election requirements.
- General cybersecurity topic-based class projects cover many security and cryptographic primitives such as user authentication, encryption and decryption, digital signatures, n-party secret sharing, and secure multiple party multiplication. Moreover, different authentication systems such as user name/password authentication or biometrics-based authentication can be designed and implemented. These class projects are independent from E-voting systems and can be used in any security course.

Students will be assigned projects of different types at different times. The assignments will reflect different student-centered interactions as follows. **Interaction between students** (i.e., their implemented software modules) and **the system**: students finish the projects and plug the implemented modules into the system to test their interaction with the rest of the system as well as the integrated system.

Student-centered interactions of class projects. These assignments will reflect different student-centered interactions. As students finish their implementations of software modules, they plug them into the system to test their interaction with the rest of the system. This requires interactions among students. Here the cut and paste method is used: cut the original standard module and put in the implemented one. This also requires the group implementing a module to interact with other groups implementing and testing other modules. This will give students hands-on experience with security principles, protocols, and systems as well as system integration, message passing, and peer-to-peer protocols. Six concrete student-centered interactions **InterA-1** to **InterA-6** are discussed in the next subsection.

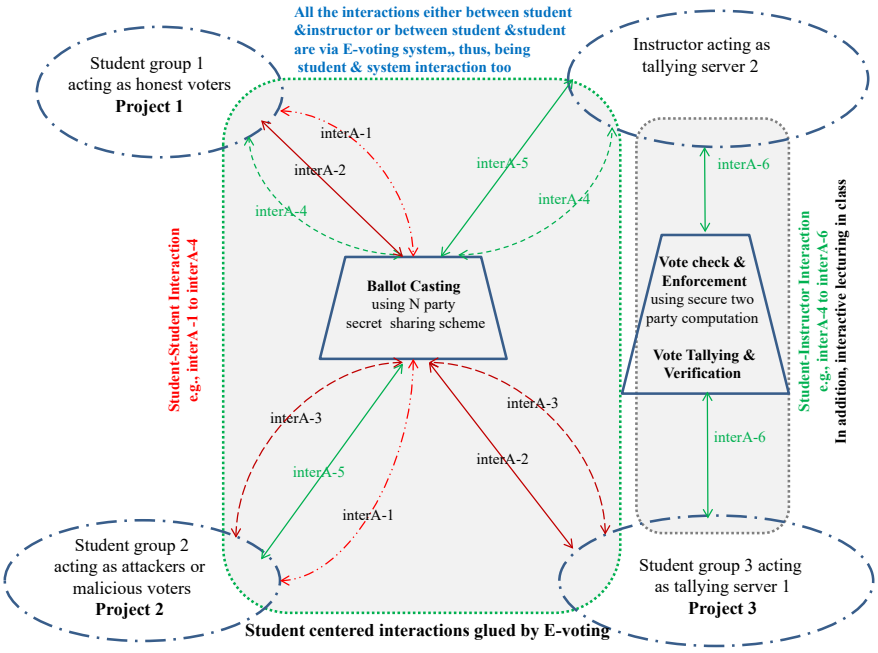

Fig. 2: Student-centered interactive class projects–students will switch groups.

Systematic design of interactive class projects. Figure 2 shows possible projects, their corresponding student groups, and the interaction among the students and the E-voting system. There are four primary types of class projects and six interactions. The instructor acts as one tallying server. Students are divided into groups. Each group plays one of the roles of honest voter, dishonest voter (attacker), and the second tallying server. During the term, student groups will switch roles so each group will implement different projects. The E-voting system connects all projects and controls all interactions.

– **Project 1**: (honest) vote casting. This type of project implements a voting process, which includes two main modules: voter authentication and forming and casting ballots. The goal is to have students consolidate their knowledge of authentication, cryptosystems, data integrity, digital signatures, and n party secret sharing. It can also include determining acceptable (legal) methods of authentication and keeping track of eligible voters. The project can be implemented at different levels of security and protection strengths: e.g., with authentication or not, encryption or not, integrity or not, digital signature or not, commitments or not. As an additional benefit, the class project also serves to educate students on the importance and difficulty of robust and bug-free implementation.
– **Project 2**: (dishonest) vote casting and attacks. This project has students implement various attacks on the E-voting system. Attacks can be designed to bypass authentication, disrupt the n party secret sharing protocol, and modify data in transition — any type of disruption or comoromise. Examples of attacks include sending invalid secret

shares, publishing an invalid commitment, and casting invalid votes such as multiple votes or unauthorized votes. The goal here is to have students not only use and consolidate the above knowledge base, but also be able to analyze and exploit the system vulnerability and to design attacks.

- **Project 3**: vote checking. This project implements one of two tallying servers. The tallying server resulting from this project and the tallying server controlled by the instructor will jointly perform vote checking. Students will extend their knowledge in areas such as homomorphic encryption and secure multi-party computation.
- **Project 4**: requirements analysis and verification. This project analyzes aspects of the E-voting system to determine if it meets specific requirements. For example, are there problems with the software that could lead to a denial of service? What assumptions does this scheme make, and are those assumptions viable in the election jurisdiction of the class (or of some other jurisdiction)? If not, what would need to change? Or, what requirements must the jurisdiction change in order to ensure this system is usable, available during the election, and meets the laws and regulations of the jurisdiction?

This leads to six possible interactions.

- **InterA-1:** *Group 1 and Group 2 students* interact using n party secret sharing to cast their ballots and to make the E-voting process run to completion. The former will cast their ballot honestly and the latter dishonestly to attempt to invalidate or disrupt the voting process.
- **InterA-2:** *Group 1 and Group 3 students*: Group 3 acts as the second tallying server. Group 1 sends their information to Group 3.
- **InterA-3:** *Group 2 and Group 3 students*: Similarly, Group 2 send information to Group 3, but they can send wrong shares, wrong ballots, or wrong commitments, or even send nothing.
- **InterA-4:** *Group 1 students and the instructor*: The students send their shares and commitment to the instructor for enforcement.
- **InterA-5:** *Group 2 students and the instructor*: As InterA-4, but with Group 2 and not Group 1.
- **InterA-6:** *Group 3 students and the instructor* interact using homomorphic encryption-based secure two party multiplication to jointly check and constrain a voter's behavior.

Design of cybersecurity topic-based class projects. The above projects are E-voting related. However, topics in information security are not limited to E-voting systems. Therefore, additional projects can use material which are generic and applicable to all courses without tying them to the E-voting system. For example, one project can have students implement different encryption algorithms such as AES and RSA with varying key lengths, independent of the E-voting system. This will help the students understand different cryptosystems, evaluate their strengths and weaknesses, and the strength and impact of different key lengths. Attacker and defender projects can impose different constraints on the tools and defenses to be used to mimic different environments of the defender.

Table 1: Projects, their modules and involved interactions

Projects	**Modules**									**Interactions**					
	1	2	3	4	5	6	7	8	9	A-1	A-2	A-3	A-4	A-5	A-6
Project 1	√	√	√		√	√		√		√	√			√	
Project 2	√	√	√		√	√	√			√		√	√		
Project 3	√	√	√	√		√	√	√	√		√	√			√
Project 4	√	√	√			√			√.	√		√	√	√	√

Table 2: Four interactive modules and their features

Module Formats	**Topic related?**	**Each topic?**	**Next activity**	**In or outside class**
Interactive lecturing	N	Y	Interactive class project	In
Interactive class project	Y	Y	Interactive self study and evaluation or Interactive evaluation	Outside
Interactive self study and evaluation	N	Y or N	Interactive evaluation	Both
Interactive evaluation	N	Y	Interactive Lecturing	Both

4.3 Interactive self-study and evaluation

These activities occur among students without the involvement of the instructor. This student-student interaction is feasible because of the unique feature of two tallying authorities having conflicting interests. The E-voting system allows students to act as one tallying authority and engage in discussion among themselves. For example, one student, acting as a tallying server, can post questions or quizzes and other students can cast votes as their answers. The results can guide students into further discussion on the related topics. This kind of student engagement and discussion may be held outside the class and would be supported by the system under study.

Anonymous evaluation by students is also useful when a project team finishes a project. In many situations, one member of the team does not put in effort on a project commensurate with the other team members, but because the team is graded as a whole, everyone gets the same grade. To ensure all team members contribute to the project and earn credit proportional to their respective contributions, the anonymous E-voting system can be used to report when one of the members is not contributing; the instructor can then decide how to proceed. Such an anonymous project evaluation mechanism can potentially impact students' involvement and contribution to team projects.

4.4 Interactive topic/class evaluation

This interactive activity occurs between students and the instructor normally at or near the end of the class. In fact, it can occur whenever a module is done or the instructor deems it necessary. For example, once a topic and its corresponding projects are finished, an immediate class evaluation on teaching of this can be done. Two types of

Table 3: Interactive framework, modules, and their application/adaption to different courses

Framework/Modules	**Generic/Topic-wise**	**Cryptography**	**Security-oriented courses**	**non-security courses**
Whole framework	Topic-wise cyclic model	Directly use	Directly use	Directly use
Interactive lecturing	General function/generic question	Directly use	Directly use	Directly use
	Topic based questions	Use all	Use some create new ones	Create all new ones
Interactive class project	Topic based projects	Use all	Use some create new ones	Create all new ones
Interactive self study and evaluation	General function	Directly use	Directly use	Directly use
	Topic based evaluation	Use all	Use some create new ones	Create all new ones
Interactive evaluation	General function/class evaluation	Directly use	Directly use	Directly use
	Topic based evaluation	Use all	Use some create new ones	Create all new ones

The general functions include instructor sets questions, students respond, students set questions, instructor sets topics/class schedules, etc.

evaluations can be conducted: topic based evaluation and general student survey. The former can use quizzes to evaluate and will guide the instructor how to proceed (i.e., advance or repeat) in a timely manner. In addition, a general class survey should be done to give students opportunities to express their opinions, including questions like: how do you feel about the current class pace? Should the class advance to the next topic? How do you feel about the instructor's knowledge on this topic? How do you feel about the instructor's enthusiasm and effort on the topic?

This module format can be used for all topics. Frequent evaluation (in an appropriate frequency) and timely feedback, as compared to a single end-of-semester evaluation practice, will improve the instruction quality and enhance the student learning outcomes.

4.5 Summary of topics, projects, and interactive modules

The modules and interactions that each project covers are summarized in Table 1. As evident, each of the four projects covers four basic cybersecurity topics: authentication, confidentiality, integrity, and access control/authorization. They together cover all topics and all interactions, and can do so from different disciplinary points of view. Depending on (types and levels) of courses, one or multiple class projects can be assigned for students to do.

Table 2 summarizes four (format/constructions of) interactive modules, their features and interconnection and transition. As can be seen from the table, within the four modules, only interactive class projects are strongly topic dependent. All the others can be adapted easily to other topics and thus, other courses.

Even though the interactive framework is designed based on E-voting technology, it can be independently applied to other security courses. The overall framework, i.e., topic based cyclic interactive learning process (interconnected by four interactive modules) can even be adapted to non-security courses. Table 3 gives a condensed overview for adapting the four module formats to other courses. For example, a non-security course would need to create new interactive class projects whereas a non-electronic voting security course may find relevance in E-voting projects.

5 Conclusions

This paper proposed a flexible cybersecurity curriculum development framework based on an E-voting system. It is configurable to different topics such as cryptography, information security, and network security course curricula. It includes modules on security-related topics that can be used in non-security-specific courses. Using this material, instructors can entice students' interest in cybersecurity and enable students to learn cybersecurity in an attractive and engaged manner.

Many future works can, or to say should, be done with the proposed new cybersecurity curriculum, including, but not limited to, developing a system to support/facilitate such a new teaching and learning methodology and testing and evaluating the effectiveness and impact of such a new curriculum.

Acknowledgement

This material is based upon work supported by the National Science Foundation under Grant Nos. DGE-2011117 and DGE-2011175. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

1. M. Bishop and D. A. Frincke, "Achieving learning objectives through e-voting case studies.," *IEEE Security and Privacy*, vol. 5, no. 1, pp. 53 – 56, 2007.
2. D. A. Kolb, "Experiential learning: Experience as the source of learning and development," *Englewood Cliffs, NJ, Prentice Hall*, 1984.
3. D. Laurillard, "Rethinking university teaching: a conversational framework for the effective use of learning technology, 2nd edition," *London, RoutledgeFarmer*, 2002.
4. G. E. Kennedy and Q. I. Cutts, "The association between students' use of an electronic voting system and their learning outcomes," *Journal of Computer Assisted Learning*, vol. 21, no. 4, pp. 260 – 268, 2005.
5. Q. I. Cutts and G. E. Kennedy, "Connecting learning environments using electronic voting systems," *Australiasian Computing Education Conference*, pp. 181–186, 2005.
6. J. R. Stowell and J. M. Nelson, "Benefits of electronic audience response systems on student participation, learning, and emotion," *Teaching of Psychology*, vol. 34, no. 4, pp. 253–258, 2007.
7. X. Zou, H. Li, Y. Sui, W. Peng, and F. Li, "Assurable, transparent, and mutual restraining e-voting involving multiple conflicting parties," in *Proceedings of the 2014 IEEE Conference on Computer Communications*, IEEE INFOCOM 2014, (Piscataway, NJ, USA), pp. 136–144, IEEE, Apr. 2014.
8. Joint Task Force on Cybersecurity Education, "Cybersecurity curricula 2017," technical report, ACM, New York, NY, USA, Dec. 2017.
9. W. Du and R. Wang, "Seed: A suite of instructional laboratories for computer security education," *ACM Journal on Educational Resources in Computing*, vol. 8, pp. 3:1–3:24, Mar. 2008.
10. W. Du, *Computer Security: A Hands-On Approach*. Seattle, WA, USA: CreateSpace, 2017.

11. J. Van Niekerk and R. von Solms, "Using bloom's taxonomy for information security education," in *Proceedings of the Sixth World Conference on Information Security Education*, vol. 406 of *IFIP Advances in Information and Communication Technology*, (Berlin, Germany), pp. 280–287, Springer, July 2009.
12. T. Flushman, M. Gondree, and Z. N. J. Peterson, "This is not a game: Early observations on using alternate reality games for teaching security concepts to first-year undergraduates," in *Proceedings of the Eighth Workshop on Cyber Security Experimentation and Test*, (Berkeley, CA, USA), USENIX Association, Aug. 2015.
13. Z. C. Schreuders and E. Butterfield, "Gamification for teaching and learning computer security in higher education," in *Proceedings of the 2016 USENIX Workshop on Advances in Security Education*, (Berkeley, CA, USA), USENIX Association, Aug. 2016.
14. J. Vykopal and M. Barták, "On the design of security games: From frustrating to engaging learning," in *Proceedings of the 2016 USENIX Workshop on Advances in Security Education*, (Berkeley, CA, USA), USENIX Association, Aug. 2016.
15. D. L. Burley and M. Bishop, "Summit on education in secure software final report," Technical Report GW-CSPRI-2011-7, The George Washington University, Washington, DC, USA, June 2011.
16. S. Raina, B. Taylor, and S. Kaza, "Security injections 2.0: Increasing engagement and faculty adoption using enhanced secure coding modules for lower-level programming courses," in *Proceedings of the Ninth IFIP World Conference on Information Security Education* (M. Bishop and N. Miloslavskaya, eds.), vol. 453 of *IFIP Advances in Information and Communication Technology*, pp. 64–74, Springer, May 2015.
17. M. Bishop and C. Elliott, "Robust programming by example," in *Proceedings of the 2009 IFIP World Conference on Information Security Education*, vol. 406 of *IFIP Advances in Information and Communication Technology*, (Berlin, Germany), pp. 140–147, Springer, June 2013.
18. M. Bishop, M. Dark, L. Futcher, J. Van Niekerk, I. Ngabeki, S. Bose, and M. Zhu, "Learning principles and the secure programming clinic," in *Proceedings of the 12th IFIP WG 11.8 World Conference on Information Security Education* (L. Drevin and M. Theocharidou, eds.), vol. 557 of *IFIP Advances in Information and Communication Technology*, (Cham, Switzerland), pp. 16–29, Springer Nature, June 2019.
19. T. Chothia and J. de Ruiter, "Learning from others' mistakes: Penetration testing iot devices in the classroom," in *Proceedings of the 2016 USENIX Workshop on Advances in Security Education*, (Berkeley, CA, USA), USENIX Association, Aug. 2016.
20. D. Formby, M. Rad, and R. Beyah, "Lowering the barriers to industrial control system security with grfics," in *Proceedings of the 2018 USENIX Workshop on Advances in Security Education*, (Berkeley, CA, USA), USENIX Association, Aug. 2018.
21. E. Sitnikova, E. Foo, and R. B. Vaughn, "The power of hands-on exercises in scada cyber security education," in *Proceedings of the Eighth IFIP World Conference on Information Security Education* (R. C. Dodge Jr. and L. Futcher, eds.), vol. 406 of *IFIP Advances in Information and Communication Technology*, (Berlin, Germany), pp. 83–94, Springer, July 2008.
22. E. Atwater, C. Bocovich, U. Hengartner, and I. Goldberg, "Live lesson: Netsim: Network simulation and hacking for high schoolers," in *Proceedings of the 2017 USENIX Workshop on Advances in Security Education*, (Berkeley, CA, USA), USENIX Association, Aug. 2017.
23. R. Weiss, J. Mache, and M. Locasto, "Live lesson: The edurange framework and a movie-themed exercise in network reconnaissance," in *Proceedings of the 2017 USENIX Workshop on Advances in Security Education*, (Berkeley, CA, USA), USENIX Association, Aug. 2017.
24. M. I. Shamos, "Electronic voting," *http:http:euro.ecom.cmu.eduprogramcoursestcr17-803*, 2014.

25. A. J. Halderman, "secure digital democracy," *https:www.coursera.orginstructorjhalderm*, 2014.
26. X. Zou, H. Li, F. Li, W. Peng, and Y. Sui, "Transparent, auditable, and stepwise verifiable online e-voting enabling an open and fair election," *Cryptography, MDPI*, vol. 1, no. 2, pp. 1–29, 2017.
27. M. Clarkson, S. Chong, and A. Myers, "Civitas: Toward a secure voting system," in *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, pp. 354–368, 2008.
28. R. W. Gardner, S. Garera, and A. D. Rubin, "Coercion resistant end-to-end voting," in *In 13th International Conference on Financial Cryptography and Data Security*, 2009.
29. J. Benaloh and A. Tuinstra, "Receipt-free secret-ballot elections," in *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, (New York), pp. 544–553, ACM, 1994.
30. K. Sako and J. Kilian, "Receipt-free mix-type voting scheme," in *Advances in Cryptology-UROCRYPT'95*, pp. 393–403, Springer, 1995.
31. L. J. Osterweil, M. Bishop, H. M. Conboy, H. Phan, B. I. Simidchieva, G. S. Avrunin, L. A. Clarke, and S. Peisert, "Iterative analysis to improve key properties of critical human-intensive processes: An election security example," *ACM Transactions on Privacy and Security*, vol. 20, pp. 5:1–5:31, Mar. 2017.
32. C. A. Neff, "A verifiable secret shuffle and its application to e-voting," in *Proceedings of the 8th ACM conference on Computer and Communications Security*, CCS '01, (New York, NY, USA), pp. 116–125, ACM, 2001.
33. D. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Comm. of the ACM*, vol. 24, pp. 84–90, Feb. 1981.
34. D. Chaum, P. Ryan, and S. Schneider, "A practical voter-verifiable election scheme," in *Proc. of the 10th European Conf. on Research in Comp. Security*, ESORICS'05, (Milan, Italy), pp. 118–139, 2005.
35. B. Lee, C. Boyd, E. Dawson, K. Kim, J. Yang, and S. Yoo, "Providing receipt-freeness in mixnet-based voting protocols," in *Proc. of Info. Security and Cryptology*, vol. 2971, pp. 245–258, 2003.
36. S. Weber, "A coercion-resistant cryptographic voting protocol - evaluation and prototype implementation," *Master's thesis, Darmstadt University of Technology*, 2006.
37. G. Brassard, D. Chaum, and C. Crepeau, "Minimum disclosure proofs of knowledge," *Journal of Computer and System Sciences*, vol. 37, pp. 156–189, 1988.
38. M. Naor, "Bit commitment using pseudorandomness," *Journal of Cryptology*, vol. 4, no. 2, pp. 151–158, 1991.
39. O. Goldreich, S. Micali, and A. Wigderson, "Proofs that yield nothing but their validity, or all languages in np have zero-knowledge proof systems," *Journal of the ACM*, vol. 38, no. 3, pp. 690–728, 1991.
40. H. Ge, S. Y. Chau, V. E. Gonsalves, H. Li, T. Wang, X. Zou, and N. Li, "Koinonia: Verifiable e-voting with long-term privacy.," *The 2019 Annual Computer Security Applications Conference (ACSAC 2019), San Juan, Puerto Rico, December 9-13*, p. Accepted, 2019.
41. J. Benaloh and M. Yung, "Distributing the power of a government to enhance the privacy of voters," in *Proceedings of the fifth annual ACM symposium on Principles of distributed computing*, PODC '86, (New York, NY, USA), pp. 52–62, ACM, 1986.
42. J. Benaloh, *Verifiable Secret Ballot Elections*. PhD thesis, Yale University, 1987.
43. J. Benaloh and M. Yung, "Distributing the power of a government to enhance the privacy of voters," in *Proc. of the 5th annual ACM Symp. on Principles of distributed computing*, PODC '86, pp. 52–62, 1986.
44. B. Schoenmakers, "A simple publicly verifiable secret sharing scheme and its application to electronic," in *Proc. of the 19th Annual Int. Cryptology Conf. on Advances in Cryptology*, CRYPTO '99, (London, UK), pp. 148–164, 1999.
45. —, *California Elections Code 2015*. Irvine, CA, USA: DFM Associates, 2015.