

Beyond the Login Screen: Teaching Full-Stack Defense-in-Depth from Software to Hardware Logic

Md Mufassal Ahmad¹, Mary Nusrat², Gahangir Hossain³

Department of Electrical Engineering, University of North Texas, Denton, Texas, USA¹

Department of Information Science, University of North Texas, Denton, Texas, USA²

Department of Data Science, University of North Texas, Denton, Texas, USA³

Abstract—Defense-in-Depth (DiD) holds that a system should be protected by multiple independent layers of security, so that the failure or compromise of a single countermeasure does not result in total system compromise. Secondary cybersecurity instruction, however, often presents security as a list of disconnected rules. This study presents the design of an instructional framework that teaches DiD to high-school students as a transferable form of systems thinking. The framework is developed within a teacher professional-development project and comprises five stages: Hook, Layer Mapping, Threat Modeling, Hardware Case Study, and Design Challenge, each grounded in an established learning theory and tied to six measurable, Bloom-aligned learning objectives. Threat modeling is introduced through a four-question process supported by student-facing adaptation of the six STRIDE categories and simple attack trees. A deliberately compact case study of the six-layer defense of logic-locked integrated circuits serves as a far-transfer task rather than as a hardware tutorial. The framework includes five assessment instruments, a five-criterion, four-level analytic rubric (5–20 points), and an implementation guide that fits the module into seven to nine 45–50 minute class periods. The contribution is primarily design-oriented: specifying three hypotheses, associated metrics, and a pretest–posttest protocol for a planned classroom pilot. By connecting familiar security protections with hardware security, DiD shifts the student’s question from “Is the system secure?” to “How does the system remain resilient when an individual protection fails?”.

Keywords—Defense-in-depth; cybersecurity education; secondary education; threat modeling; STRIDE; instructional design; hardware security; logic locking

I. INTRODUCTION

High-school students use digital systems every day and are exposed to the corresponding risks. Weak or reused passwords compromise personal accounts, phishing messages steal credentials, unpatched devices spread malware, and insecure networks leak sensitive information. Schools themselves are targets: the learning management system used by many districts was recently breached twice within a short period [1]. Cybersecurity education therefore needs to move beyond teaching security as a set of discrete actions or tools. Students need to understand security as a system in which several measures combine to reduce risk [2], [3].

Defense-in-Depth (DiD) therefore provides an appropriate organizing concept for cybersecurity instruction. It assumes that no single countermeasure can effectively address all possible threats [4]. For example, a strong password may reduce the

likelihood of successful guessing attacks, multi-factor authentication may limit the consequences of compromised credentials, and login alerts may help detect unauthorized access attempts. Each mechanism serves a distinct defensive function, and their combined use provides greater resilience than reliance on any individual safeguard alone. The same principle applies to more advanced security domains. Rahman *et al.* [5] demonstrate that protecting the secret key of a logic-locked integrated circuit (IC) requires multiple independent safeguards against threats such as reverse engineering, optical probing, electrical probing, scan-chain attacks, and oracle attacks. Although the underlying techniques are technically complex, the core principle remains straightforward: valuable assets require multiple independent layers of protection because adversaries may exploit different attack paths.

For secondary education, DiD offers three key advantages. First, it connects directly to students’ everyday use of technology. Second, it promotes systems thinking by requiring students to identify assets, threats, vulnerabilities, countermeasures, and potential failure paths. Third, it bridges familiar cybersecurity practices with engineering-oriented security design, enabling teachers to present cybersecurity as a structured reasoning process rather than as a checklist of isolated rules.

A. Scope and Addressing Design Gaps

This work is an instructional-design study. Following the design-based research tradition, in which a theoretically grounded intervention is specified before it is evaluated in iterative classroom cycles [6]. The study documents the design phase and is organized around three design questions:

- **DQ1.** How can DiD be decomposed into a sequence of learning activities that is accessible to students with no prior security training?
- **DQ2.** How can an advanced hardware-security case be reduced so that it supports transfer of DiD reasoning instead of becoming a technical tutorial?
- **DQ3.** Which instruments and criteria make DiD reasoning—asset identification, threat identification, layer diversity, and failure-path analysis, observable and assessable by a classroom teacher?

The hypotheses that follow from this design, and the protocol for testing them, are stated in Section VIII.

B. Contributions

- Six measurable learning objectives aligned with Bloom taxonomy and mapped to stages and instruments (Table I, Fig. 4).
- A five-stage instructional sequence for DiD in which every stage is justified by an established learning theory (Table II, Fig. 3).
- A student-facing adaptation of STRIDE and attack trees that gives the threat-modeling stage methodological depth (Table III, Fig. 4).
- A compact logic-locking case study used as a far-transfer task (Table IV, Fig. 5).
- An assessment plan with an analytic rubric and performance criteria (Tables V and VI), and an implementation guide covering duration, prerequisites, preparation, and resources (Table VII).

The remainder of the study is organized as follows. Section II reviews related work and the learning theories that ground the design. Section III defines DiD and explains why it suits secondary learners. Section IV describes the instructional context and the learning objectives. Section V presents the five-stage framework. Section VI presents assessment, Section VII addresses classroom implementation, Section VIII states limitations and the evaluation plan, and Section IX concludes the work.

II. RELATED WORK AND THEORETICAL FOUNDATIONS

A. Cybersecurity Education in K-12

A growing body of work argues for integrating cybersecurity into K-12 curricula. Pencheva *et al.* [3] argue that cybersecurity should be woven into secondary education to foster a security mindset, and note that many programs present security as a collection of rules. Kamaludeen *et al.* [7] propose a cyber-protection framework for the K-12 sector that spans technical, administrative, and behavioral defenses. Vajpayee and Hossain [8] examine authentic assessment in high-school cybersecurity and stress that students must learn to evaluate cyber assets and risks holistically. Recent systematic reviews confirm that the field is active but uneven. Ibrahim *et al.* [9] find that most aspects of cybersecurity are not taught systematically at K-12 level; Sağlam *et al.* [11] report that content varies widely between countries and leaves gap; Arishi *et al.* [12] identify games, mobile applications, and tailored programs as effective awareness practices; and Xu and Li [13] map trends, challenges, and opportunities for younger learners. Across these reviews the dominant topics are awareness and safe behavior. Layered-defense reasoning is rarely an explicit learning goal.

B. Teaching Threat Modeling

Threat modeling is well developed in professional practice. STRIDE classifies threats into six categories [15], attack trees decompose an attacker's goal into alternative paths [16], PASTA defines a seven-stage risk-centric process [17], and surveys compare these and other methods [18]. Educational

use has been studied mainly at university level. Joshi *et al.* [19] found that software-engineering students who received systems-thinking instruction together with STRIDE identified system-level as well as component-level threats, whereas students taught STRIDE alone concentrated on components. That result supports the premise of this study, that threat categories should be taught inside a systems view, but no comparable adaptation exists for secondary learners.

C. Learning Theories Informing the Design

Five strands of learning theory inform the proposed framework. *Constructivism* holds that learners build new understanding on prior knowledge, and that learning is most productive within the zone of proximal development, where support from a teacher or peer is available [20]. *Inquiry-based learning* organizes instruction as a cycle of orientation, conceptualization, investigation, conclusion, and discussion [21]. *Cognitive apprenticeship* makes expert reasoning visible through modeling, coaching, scaffolding, and fading [22], [23]. *Experiential learning* describes a cycle of concrete experience, reflective observation, abstract conceptualization, and active experimentation [24]. Research on *transfer learning* shows that learners abstract a general schema more reliably when they compare two analogous cases than when they study one [25], [26]. Section V assigns each stage to the theory that motivates it, and the revised Bloom taxonomy [27], [28] is used to state objectives and to order the stages by cognitive demand [29].

D. The Gap

Existing K-12 research establishes the importance of layered-defense thinking; however, it does not provide a scaffolded, theory-grounded instructional sequence with clearly assessable outcomes. While existing threat-modeling pedagogy primarily targets university students, this work addresses both gaps and incorporates a hardware-security case to examine whether DiD reasoning can transfer to an unfamiliar system.

III. DEFENSE-IN-DEPTH FOR SECONDARY LEARNERS

A. Principle

DiD deploys multiple independent layers of security to reduce the risk that the failure of one component compromises the entire system. The concept originates from military strategy, in which successive defensive lines delay, weaken, and contain an adversary, and has since become a widely adopted principle in information and industrial control system security [4], [30], [31]. Each layer is designed to prevent, detect, contain, or facilitate recovery from specific categories of attack. A strong password resists guessing but cannot stop phishing or malware. Similarly, a firewall can block many network-based attacks but cannot prevent a user from entering credentials into a fraudulent website.

An essential distinction between *redundancy* and *diversity*, is that redundant layers repeat a similar protection, whereas diverse layers address different attack vectors [4], [5]. Students often assume that a strong password means a stronger system. A system that has only password security, with no added recovery measures, no device security, and no user awareness, remains fragile however strong the password is. Fig. 1 contrasts the single vs multi-layered defense configurations.

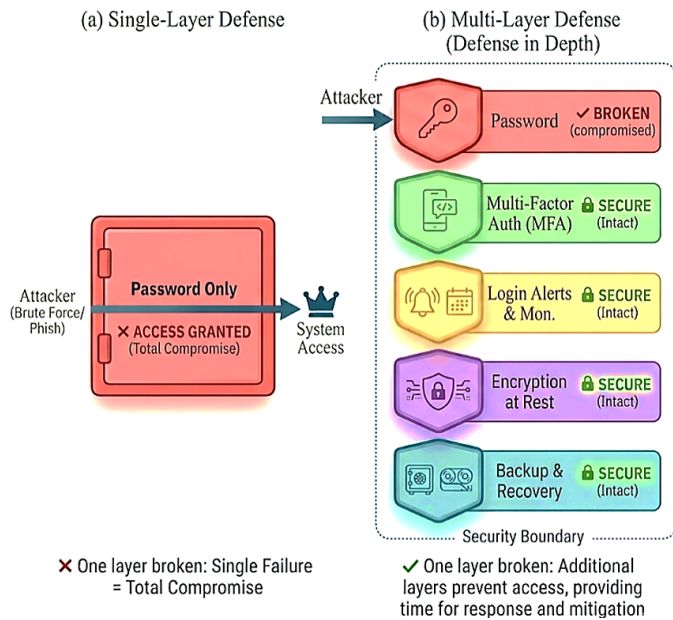


Fig. 1. Single-layer defense versus multi-layered defense.

B. Rationale for the Secondary Level

Secondary-level students increasingly rely on information systems for communication, learning, financial activities, social interactions and entertainment [7], [8], however, cybersecurity is often encountered as a set of disconnected rules rather than as an integrated component of their digital activities [10]. The DiD instructional approach targets three primary outcomes. The first outcome is *awareness*: students who understand that successful cyberattacks often exploit a single unprotected point may be more motivated to adopt multiple protective measures. The second is *systems thinking*: DiD requires students to identify what must be protected, how it could be compromised, which safeguards are already in place, and where gaps remain. The third is *transfer*: the layered reasoning developed through familiar contexts, such as email accounts, can be extended to more complex environments, including enterprise systems, critical infrastructure, and hardware. DiD is well suited to this age group because it does not require advanced technical skills [2] and can be taught through scenarios that are familiar to students.

C. Everyday Examples used in Instruction

Three familiar systems supply the examples used in Stages I to III.

1) *Personal accounts*: A student who reuses one password relies on a single layer. A layered alternative combines a unique password held in a password manager, multi-factor authentication (MFA), login alerts, securely stored recovery codes, and phishing awareness. Each layer answers a different threat: guessing, stolen credentials, undetected intrusion, loss of access, and deception.

2) *School networks and devices*: Schools combine account controls, web filtering, antivirus software, patching, network segmentation, backups, and monitoring [7], [31]. Patching

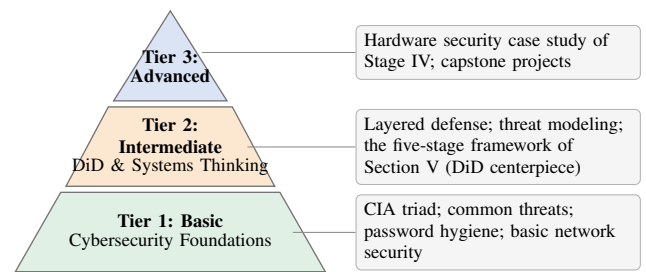


Fig. 2. Three-tier cybersecurity training for STEM teachers in the PERFECT project; the DiD framework forms Tier 2 and its capstone case belongs to Tier 3.

closes vulnerabilities, antivirus detects malicious files, segmentation contains an infection, backups enable recovery from ransomware, and monitoring reveals unusual activity. The example shows that security is organizational as well as individual, and that prevention, detection, containment, and recovery are different functions.

3) *Human behavior*: Students often perceive that cybersecurity is purely technical domain [3]. However, cyberattacks frequently exploit human factors such as trust, fear, curiosity, or ignorance, and a technically sound system fails when users unintentionally bypass its protections [14]. Accordingly, the module emphasizes the user as an integral layer of defense rather than as an entity external to the security system.

IV. INSTRUCTIONAL CONTEXT AND LEARNING OBJECTIVES

A. The PERFECT Project

The framework was developed within PERFECT (Providing Environment and Resources for STEM 9th–12th Graders in Effective Cybersecurity Training), a project that prepares high-school STEM teachers to teach cybersecurity [32], [33]. The project follows a train-the-trainer model with three tiers, shown in Fig. 2. Tier 1 covers foundational concept such as the confidentiality-integrity-availability triad, common cyber-threats, and password hygiene. Tier 2 covers DiD and systems thinking, and the five-stage framework presented in this work. Tier 3 covers advanced case studies, including the hardware case in Stage IV. The framework therefore has a dual use: it is a teacher-training module and, with minor adaptation, a lesson sequence for the high-school classroom.

B. Learning Objectives

Table I presents the six learning objectives (LOs), begins with an observable verb drawn from the revised Bloom's taxonomy [28] and is assessed through at least one instrument described in Section VI. The objectives are aligned with the Grade 9-12 cybersecurity expectations of the CSTA standards, which emphasize recommending and comparing appropriate security measures for specific scenarios [34], [35], as well as with the NICE's framework emphasis on task-oriented cybersecurity competencies [36].

TABLE I. MEASURABLE LEARNING OBJECTIVES

LO	On completing the module, students can...	Bloom level	Stage
1	Explain DiD and why a single layer is a single point of failure	Understand	I
2	Classify the protections of a familiar system by the threat addressed and by function (prevent, detect, contain, recover)	Analyze	II
3	Construct a threat model using the four questions, the STRIDE prompts, and an attack tree	Apply, Analyze	III
4	Match the six IC defense layers to the attacks they stop and to everyday analogues	Analyze (transfer)	IV
5	Design a layered defense plan for a system of their choice	Create	V
6	Evaluate a defense plan for diversity versus redundancy and trace its failure paths	Evaluate	III, V

V. THE FIVE-STAGE INSTRUCTIONAL FRAMEWORK

The framework moves students from intuition, through structured analysis, to independent design. Fig. 3 shows the five stage progression, and Table II summarizes, for every stage, the guiding question, the DiD principle, the student output, and the learning theory that motivates the stage. Cognitive demand rises from *understand* to *create* [28], and teacher support is withdrawn gradually, as cognitive apprenticeship prescribed in stages [22].

A. Stage I: Hook and Motivation

Stage I introduces DiD through relatable failures, for example an account taken over through a weak password, a school computer infected through an email attachment, or a ransomware incident that began on one unpatched machine [8]. No security vocabulary is required. The instructional goal is to make the fragility of single-layer protection visible, so that students leave with a question of their own: many failures occur not because every protection failed, but because the system depended on one. Stage II then turns from failures to the protections that students already have.

B. Stage II: Layer Mapping

Students identify and document the security measures implemented across systems they regularly use, including smartphones, school accounts, laptops, cloud storage services, and home Wi-Fi networks [7], [37]. Common examples include passwords, biometric authentication, multi-factor authentication (MFA), software updates, antivirus software, login notifications, data backups, and permission settings. Students then classify each security measure according to two dimensions: the threat it is intended to address and its defensive function—prevention, detection, containment, or recovery. This dual classification helps learners to recognize that security layers are not interchangeable, as each serves a distinct defensive purpose. It also allows students to identify potential imbalances in their existing security practices, particularly with an emphasis on preventive measures relative to detection and recovery mechanisms. However, this inventory is organized primarily from the defender's perspective by focusing on existing safeguards, whereas an attacker typically approaches a system by identifying exploitable assets, vulnerabilities, and weaknesses. This distinction provides the motivation for Stage III, which introduces threat-oriented analysis.

C. Stage III: Threat Modeling

Stage III shifts the perspective from the defender to the attacker by introducing a threat-oriented analysis. Students examine a selected system by addressing four fundamental principles of professional threat modeling [15]:

- What asset requires protection?
- What threats could compromise the asset?
- What countermeasures are currently in place to mitigate these threats?
- What security gaps or single points of failure remain?

1) *Choice of method*: Without structured guidance, novice learners may identify only a limited set of familiar attacks when addressing the second principle. An established threat-modeling method can provide a systematic framework for identifying a broader range of threats. Four approaches were considered. PASTA follows a seven-stage, risk-centric process that typically requires organizational and business context beyond the experience of novice learners [17]. DREAD provides quantitative risk scores, but the subjectivity associated with assigning these scores has been identified as a limitation [18]. STRIDE, in contrast, organizes threats into six mnemonic categories [15] that can be accessible concept for learners, while attack trees [16] provide a visual representation of alternative attack paths and potential points of failure. STRIDE is selected as the primary framework for systematically identifying relevant threats, while attack trees were incorporated to examine security gaps and potential single points of failure. To maintain accessibility for secondary-level learners, risk evaluation is simplified to qualitative high, medium, or low ratings based on the estimated likelihood and potential impact of each threat [38].

2) *STRIDE prompts*: Table III presents the student-facing prompt associated with each STRIDE category, an illustrative example drawn from an everyday context, and the corresponding threat identified in the hardware security case introduced in Stage IV. Rather than requiring students to memorize the STRIDE category names, the prompts serve as a structured checklist for systematically considering a broader range of potential threats beyond those that may initially come to mind. The hardware-security mappings presented in the final column were developed specifically for the proposed instructional framework and are revisited in Stage IV to demonstrate how threat-modeling concepts can be transferred from familiar cybersecurity scenarios to hardware security contexts.

3) *Attack trees*: To address the fourth principle, students construct a simplified attack tree in which the attacker's objective is represented as the root, alternative attack paths as branches, and existing security layers as controls associated with the branches they are intended to mitigate (Fig. 4). The attack tree is then used to evaluate the DiD design according to two criteria. First, any branch lacking a protective layer indicates a potential single point of failure. Second, multiple protections concentrated on the same branch reflect redundancy rather than defensive diversity. A more resilient design distributes preventive, detective, containment, and recovery mechanisms across distinct attack paths [31], [38].

Following the cognitive apprenticeship approach, the teacher first models a complete analysis, then guides groups through the analysis of a second system using the worksheet,

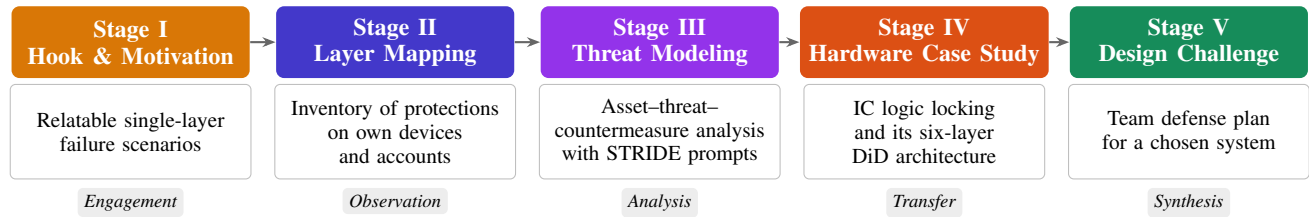


Fig. 3. Five-stage instructional framework for teaching Defense-in-Depth.

TABLE II. FIVE-STAGE FRAMEWORK: GUIDING QUESTION, PRINCIPLE, OUTPUT, AND THEORETICAL BASIS

Stage	Focus	Guiding question	DiD principle	Student output	Theoretical basis and how it is applied
I	Hook and Motivation	What happens if the only layer fails?	Single point of failure	Short reflection	Inquiry-based learning, orientation phase [21]: a puzzling failure creates the question that the module answers, and activates prior knowledge [20].
II	Layer Mapping	Which protections already exist, and what does each one do?	Independent, diverse layers	Layer inventory	Constructivism and the first half of Kolb's cycle [24]: students start from concrete experience with their own devices and reflect on it.
III	Threat Modeling	What are the asset, the threats, the countermeasures, and the gaps?	Asset-threat alignment	Threat worksheet and attack tree	Cognitive apprenticeship [22], [23]: the teacher models expert reasoning aloud, the worksheet and prompt card scaffold it, and support fades across examples.
IV	Hardware Case Study	Does the same reasoning protect an IC?	Concept transfer	Case response	Analogical transfer [25], [26]: comparing two structurally similar cases (account, ICs) leads students to abstract the DiD schema.
V	Design Challenge	How can we build a layered defense ourselves?	Applied systems thinking	Team defense plan	Experiential learning, active experimentation [24], carried out in teams so that peers provide support within the zone of proximal development [20].

TABLE III. STUDENT-FACING ADAPTATION OF STRIDE USED IN STAGE III AND REVISITED IN STAGE IV

STRIDE category	Prompt given to students	Everyday example	Typical defensive layer	Counterpart in the IC case (Stage IV)
Spoofing	Could someone pretend to be me or a service I trust?	Fake school login page	MFA; phishing awareness	Cloned or overproduced ICs sold as genuine
Tampering	Could someone change the data or the device?	Malware altering files	Patching; antivirus; backups	Hardware Trojan inserted during fabrication
Repudiation	Could someone act without leaving a trace?	Unnoticed login from a new device	Login alerts; activity logs	Probing attempt that goes undetected
Information disclosure	Could someone see what they should not?	Reused password leaked in a breach	Unique passwords; encryption	Key recovered by reverse engineering, probing, or SAT attack
Denial of service	Could someone stop me from using it?	Ransomware; lost phone	Backups; recovery codes	Trojan that disables the ICs
Elevation of privilege	Could someone obtain more access than they are allowed?	Student account gaining administrator rights	Least privilege; role-based permissions	Test (scan) access misused to read internal state

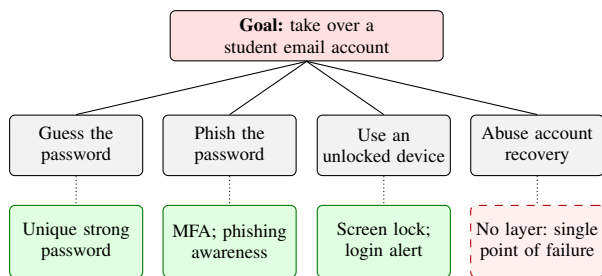


Fig. 4. Example of the simplified attack tree drawn in Stage III. Shaded boxes on the lower row are existing layers; the dashed box marks an unprotected branch.

and finally allows groups to analyze a third system using only the prompt card. At this stage, students have developed a structured method for analysis. Stage IV examines whether this method can be transferred to a different domain.

D. Stage IV: Hardware Security Case Study

The case study is intentionally concise, as its primary purpose is to facilitate knowledge transfer. Students are introduced only to the hardware concepts necessary to understand and apply the comparison.

1) *The problem:* Integrated circuits (ICs) are often designed by one organization and fabricated, packaged, and tested by external foundries and service providers, which may expose the design to potentially untrusted entities within the semiconductor supply chain. Such exposure can create risks including intellectual-property theft, unauthorized overproduction, reverse engineering, and the insertion of malicious circuitry known as hardware Trojans [5], [39]. *Logic locking* is one hardware-security technique used to protect circuit functionality and intellectual property by introducing key-controlled logic that enables the IC to operate correctly only when the appropriate secret key is applied [40], [41]. For instructional purposes, this mechanism can be compared to a machine that operates only when the correct combination is

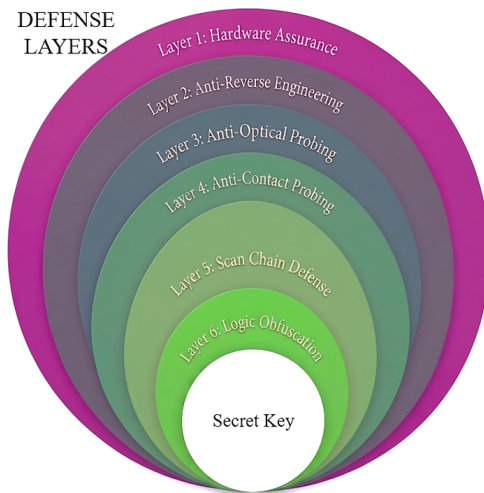


Fig. 5. Defense-in-Depth layers surrounding the secret key of a logic-locked integrated circuit (adapted from [5]).

entered. In this analogy, the secret key represents the asset that requires protection.

2) *Why one lock is insufficient:* An adversary may attempt to recover the key through multiple attack paths. These include supplying inputs to a functioning IC and inferring the key from its outputs (oracle or SAT attacks), reconstructing the circuit through physical analysis, observing internal signals optically without direct contact, probing internal wires, or exploiting scan chains intended for testing [5], [42], [43]. A defense that protects against one attack path may not prevent others. This reflects the same principle students observed with passwords in Stage I.

3) *The six layers:* Rahman *et al.* [5] arrange six defenses around the key, as in Fig. 5. Table IV pairs each layer with the attack it defeats and with an everyday analogue, and it is the only hardware material students must work with. Representative techniques are Trojan detection for hardware assurance [39], camouflaging for anti-reverse engineering [40], optical-probing detection [44], scan obfuscation [42], [43], and SAT-resistant locking such as SARLock, Anti-SAT, and SFLL [45]–[47]. Students often do not study how these techniques work.

4) *Classroom activity:* Groups receive the designated asset (the key), together with shuffled cards representing five attack paths and six defensive layers. They apply the four analytical prompts and STRIDE-based guidance introduced in Stage III, position each defensive layer on the attack path it is intended to mitigate, and compare the resulting analysis with Fig. 4 and Table III. The concluding discussion introduces risk-proportional reasoning: a personal account, a school network, and an integrated circuit may warrant different levels of security investment, although the underlying analytical process remains consistent. After applying the same method to two distinct systems, students are expected to use it independently.

E. Stage V: Design Challenge

Learners as a team apply DiD reasoning to a system of their choice, such as a student email account, a school network,

a smart-home device, an online banking portal, or a hospital database. Each team must 1) identify the asset and explain its importance, 2) list at least four threats drawn from at least three STRIDE categories, 3) propose one or more defensive layers for each threat and classify the function of each layer, and 4) explain the consequences of the failure of any single layer. The emphasis is on the rationale behind the design rather than on the technically perfect mechanisms. The teams present their plans to the class and the plans are evaluated using the rubric described in Section VI.

VI. ASSESSMENT

Assessment measure students' ability to define DiD. It determines whether learners can identify assets and threats, select diverse defensive layers, and reason about potential failures. Table V presents the five assessment instruments, the objectives it measures, and the performance criterion for each. Formative assessments evaluate understanding throughout Stages I–IV, while summative assessments determine whether students can apply DiD reasoning to a new system.

1) *Analytic rubric:* Table VI provides the rubric used for the design task and for scenario analysis. It is analytic instead of holistic, since analytic rubrics with explicit level descriptors improve scoring consistency and give students usable feedback [48], [49]. The four content criteria correspond to the four threat-modeling questions, and a fifth addresses communication. Scores range from 5 to 20. A total of at least 15 with no criterion at level 1 is taken as meeting the module objectives. Teachers may adjust the assessment weights and performance thresholds to align with assessment policies.

2) *Evaluation metrics:* For teachers and for the planned pilot, the instruments yield five quantitative indicators: the quiz score, the number of distinct STRIDE categories in a threat list, the number of defensive functions covered by a plan, the count of unprotected branches remaining in an attack tree, and the rubric with its per-criterion profile.

VII. CLASSROOM IMPLEMENTATION

A. Preparation and Resources

Table VII presents the recommended implementation plan. The complete intermediate course requires 15 modules with 120–180-minutes to complete each module. When instructional time is limited, Stages I and II can be combined into a single period, and Stage IV may be reduced to the card-based activity. All activities is designed to be conducted in learning management systems using shared digital document.

1) *Prerequisites:* Participating teachers and their students are expected to have basic digital literacy and experience. No prior background in programming, networking, or electronics is required. Teachers enrolled in intermediate module were expected to complete PERFECT Tier 1 or possess equivalent foundational knowledge. Stage IV requires only a brief introduction and does not assume any prior expertise in semiconductor technology.

2) *Classroom management and workload:* Group work in Stages III–V can be managed more effectively by assigning specific roles, such as asset owner, attacker, defender, and recorder, and rotating these roles between stages so that

TABLE IV. DEFENSE-IN-DEPTH LAYERS: HARDWARE SECURITY AND EVERYDAY ANALOGUES

Layer	Hardware defense (IC)	Attack defeated	Everyday analogue
1. Hardware assurance	Trojan detection through imaging and inspection	Trojan insertion	Verifying software authenticity; trusted sources
2. Anti-reverse engineering	Camouflaged cells; dummy vias; layout obfuscation	Reverse engineering	Code obfuscation; license enforcement
3. Anti-optical probing	Optical scattering; opaque layers; sensors	Optical (contactless) probing	Encryption of data in transit
4. Anti-contact probing	Active wire shield; probe-attempt detector	Electrical probing	Physical server security; tamper-evident seals
5. Scan-chain defense	Encrypted or dynamically obfuscated scan chains	Scan-chain attacks	MFA; role-based permissions
6. Logic obfuscation	SARLock, Anti-SAT, SFLl logic locking	SAT and oracle attacks	Strong passwords; password managers

TABLE V. ASSESSMENT INSTRUMENTS, OBJECTIVES, AND PERFORMANCE CRITERIA

Instrument	Type	LOs	Evidence and performance criterion	Weight
Concept quiz	Form.	1, 2	8–10 items on definitions, layer functions, and diversity versus redundancy; criterion: 70% correct	15%
Scenario analysis	Summ.	3, 6	Written breach narrative; student names the failed or missing layer and proposes a layer with a <i>different</i> function; rubric criteria C2–C4	20%
Hardware case response	Form.	4	Six layer–attack pairs matched and one analogy explained; criterion: five of six pairs correct	15%
Layered design task	Summ.	5, 6	Team defense plan from Stage V; full rubric (Table VI)	35%
Reflection writing	Form.	1	One personal system, one gap found, one action planned; scored for completion	15%

Note: LOs. = Learning Objectives; Form. = formative; Summ. = summative.

TABLE VI. ANALYTIC RUBRIC FOR THE DEFENSE-IN-DEPTH DESIGN TASK (5–20 POINTS)

Criterion	1 – Beginning	2 – Developing	3 – Proficient	4 – Advanced
C1. Asset identification	Asset missing, or confused with a device or tool	Asset named, but its value or owner is unclear	Asset named, with why it matters and to whom	Also separates the primary asset from supporting assets (e.g., the recovery email)
C2. Threat analysis	Fewer than two threats, or threats unrelated to the asset	Two or three threats, all from one STRIDE category	At least four specific threats from three or more STRIDE categories	Also includes human and physical threats and a realistic attacker capability
C3. Layer–threat alignment and diversity	Protections listed without links to threats	Every threat has a layer, but layers repeat one function	Every threat mapped to a layer; layers cover at least three of prevent, detect, contain, recover	Also justifies layers by asset value and cost, and removes redundant layers
C4. Failure-path reasoning	Not addressed	States that other layers “help” without tracing a path	For at least two layers, traces what the attacker gains on failure and which layer acts next	Also finds a remaining single point of failure and proposes a fix
C5. Communication of rationale	Plan cannot be followed	Understandable, but choices are not justified	Clear diagram or table with a rationale for each choice	Also uses correct vocabulary; a peer could implement the plan

TABLE VII. RECOMMENDED IMPLEMENTATION PLAN (45–50 MINUTE PERIODS)

Stage	Periods	Teacher preparation and materials	Grouping
I	1	Select two or three recent, age-appropriate incidents; prepare discussion prompts	Whole class; think–pair–share
II	1	Layer-inventory sheet; check school policy on personal devices	Pairs
III	2	Rehearse one worked example; threat worksheet, STRIDE prompt card, chart paper	Groups of 3–4
IV	1–2	Read the case brief (Fig. 5, Table IV); print attack and layer cards	Groups of 3–4
V	2–3	Plan template; rubric handed out in advance; presentation schedule	Teams of 3–4 with roles

each student has an opportunity to practice the attacker’s perspective. Preparation effort is concentrated primarily in Stages III and IV, where the teacher should be familiar with at least one worked example. Grading effort is concentrated in Stage V, where the analytic rubric enables the instructor to complete a single scoring pass for each team rather than for each individual student.

B. Pedagogical Challenges and Mitigations

1) *Abstraction*: Concepts such as threats, attack surfaces, and defensive layers can be abstract and difficult for novice

learners to visualize [3]. The module therefore begins with familiar physical analogies, such as a locked door or a vault, before progressing to cybersecurity systems, and uses visual representations to reinforce these concepts (Fig. 1, Fig. 4).

2) *Technical overload*: Logic locking, optical probing, and SAT attacks extend beyond the scope of the secondary-level curriculum. The case study is therefore limited to one figure, one table, and a card-based activity, and students are assessed on their ability to match defensive layers to corresponding attacks rather than on the underlying technical mechanisms.

3) *Differentiated skill level*: Students may enter the module with varying levels of computing experience. Teachers can therefore adjust the complexity of the selected system: beginners may analyze a personal account, while more advanced students may examine an IC or a hospital database in Stage V [7].

4) *Perceived irrelevance*: Some students may not initially perceive cybersecurity as personally relevant. Stage I therefore introduces familiar threats, such as password reuse, phishing, lost devices, and failed account recovery, to connect cybersecurity concepts with students' everyday experiences.

VIII. LIMITATIONS AND EVALUATION PLAN

The primary limitation of this work is that it presents a design framework; an empirical evaluation has yet to be conducted. Therefore, its effectiveness remains to be empirically established in implementation. The stage durations presented in Table VII and the rubric thresholds are design recommendations that may be refined based on findings from future pilot implementations. The hardware mapping in Table III is intended as an instructional aid rather than a formal threat model of logic locking. Similarly, everyday analogies in Table IV are approximated.

The next phase of this research [6] will involve a classroom pilot to examine three hypotheses:

- H1. After completing the module, learners will identify more distinct threat categories and defensive functions for a given system than before completing the module.
- H2. After completing the module, learners will analyze an unfamiliar system not used during instruction at rubric level 3 or above on criteria C2-C4.
- H3. Two teachers independently scoring the same student plans using Table VI will achieve at least moderate inter-rater agreement (Cohen's $\kappa \geq 0.6$).

The pilot study will employ a single-group pretest-posttest design using the concept quiz and a scenario-analysis task as pre- and post-measures, together with the indicators described in Section VI, student feedback, and teacher observation logs documenting the actual time required for each stage. Future work will adapt the module for introductory and advanced courses and explore interactive and gamified implementations of the five-stage framework [50].

IX. CONCLUSION

Defense-in-Depth helps learners understand security as a layered and resilient system rather than as a single protective mechanism. This study presents a five-stage framework for teaching DiD in high school. In response to DQ1, the sequence progresses from a motivating failure scenario through layer mapping and STRIDE-supported threat modeling to an independent design task, with each stage grounded in an established learning theory and aligned with measurable objectives. In response to DQ2, the logic-locking case is reduced to one concept, one table, and one card-based activity so that it functions primarily as a transfer learning task. In response to DQ3, five assessment instruments and a five-criterion analytic rubric make asset identification, threat analysis, layer diversity, and failure-path reasoning observable and assessable. The

framework has not yet been evaluated with pilot study. The intended outcome is that students learn to ask not only whether a system is protected, but also how it remains secure when an individual protective measure fails.

ACKNOWLEDGMENT

This work was supported by the United States Office of Naval Research (ONR) under Grant No. N00014-23-1-2454 through the PERFECT (Providing Environment and Resources for STEM 9th–12th Graders in Effective Cybersecurity Training) project at the University of North Texas. The authors thank the participating STEM educators from Amarillo ISD, Boyd ISD, Bushland ISD, Canyon ISD, and Denton ISD for their engagement and feedback on the course modules, and the faculty members, graduate researchers, and student assistants at the University of North Texas and West Texas A&M University who contributed to curriculum development, instructional delivery, and project coordination.

REFERENCES

- [1] Z. Whittaker, "Instructure strikes deal with hackers who breached it twice," TechCrunch, May 2026, [Online]. Available: <https://techcrunch.com/2026/05/12/instructure-strikes-deal-with-hackers-who-breached-it-twice/>.
- [2] S. P. Mohammed, G. Hossain, and S. Y. Quadri Ameen, "Cybersecurity data visualization: Designing a course for future high school students," in *Proc. 12th Int. Symp. Digital Forensics and Security (ISDFS)*, San Antonio, TX, USA, Apr. 2024, pp. 1–7.
- [3] D. Pencheva, J. Hallett, and A. Rashid, "Bringing cyber to school: Integrating cybersecurity into secondary school education," *IEEE Security & Privacy*, vol. 18, no. 2, pp. 68–74, Mar. 2020.
- [4] H. Son and S. Kim, "Defense-in-depth strategy for smart service server cyber security," in *Computer Applications for Communication, Networking, and Digital Contents*. Berlin, Germany: Springer, 2012, vol. 350, pp. 181–188.
- [5] M. T. Rahman *et al.*, "Defense-in-depth: A recipe for logic locking to prevail," *Integration*, vol. 72, pp. 39–57, May 2020.
- [6] Design-Based Research Collective, "Design-based research: An emerging paradigm for educational inquiry," *Educ. Researcher*, vol. 32, no. 1, pp. 5–8, 2003.
- [7] M. Kamaludeen, S. Ismaeel, S. Asiri, T. Allen, and C. Scarfo, "A framework for cyber protection (FCP) in K–12 education sector," *IET Conf. Proc.*, vol. 2020, no. 6, pp. 239–244, Jun. 2021.
- [8] P. Vajpayee and G. Hossain, "Cybersecurity education in high school: Exploring cyber assets, cyber value at risk, and authentic assessment," in *Proc. IEEE Frontiers in Education Conf. (FIE)*, Washington, DC, USA, Oct. 2024, pp. 1–9.
- [9] A. Ibrahim, M. McKee, L. F. Sikos, and N. F. Johnson, "A systematic review of K–12 cybersecurity education around the world," *IEEE Access*, vol. 12, pp. 59 726–59 738, 2024.
- [10] M. Nusrat, S. Bhuiyan, and G. Hossain, "Don't Just Teach, Explain! A Gamified 20Q Recommender for Cybersecurity Education," *arXiv preprint arXiv:2604.26964*, 2026.
- [11] R. B. Sağlam, V. Miller, and V. N. L. Franqueira, "A systematic literature review on cyber security education for children," *IEEE Trans. Educ.*, vol. 66, no. 3, pp. 274–286, 2023.
- [12] A. A. Arishi, N. H. Kamarudin, K. A. A. Bakar, Z. B. Shukur, and M. K. Hasan, "Cybersecurity awareness in schools: A systematic review of practices, challenges, and target audiences," *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 12, pp. 467–473, 2024.
- [13] Y. Xu and H. Li, "Cybersecurity matters for primary school students: A scoping review of the trends, challenges, and opportunities," *IEEE Trans. Learn. Technol.*, vol. 18, pp. 513–529, 2025.

- [14] M. Nusrat, S. Bhuiyan, and G. Hossain, "Securing the Cognitive Enterprise: Zero-Trust Agentic Knowledge Retention in Cyber-Physical Systems", in *Proc. 7th Int. Conf. Artificial Intelligence, Robotics, and Control (AIRC)*, Savannah, GA, USA, 2026, pp. 421–427, doi: 10.1109/AIRC69745.2026.11631472.
- [15] A. Shostack, *Threat Modeling: Designing for Security*. Indianapolis, IN, USA: Wiley, 2014.
- [16] B. Schneier, "Attack trees," *Dr. Dobbs's J.*, vol. 24, no. 12, pp. 21–29, Dec. 1999.
- [17] T. UcedaVélez and M. M. Morana, *Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis*. Hoboken, NJ, USA: Wiley, 2015.
- [18] N. Shevchenko, T. A. Chick, P. O'Riordan, T. P. Scanlon, and C. Woody, "Threat modeling: A summary of available methods," Software Engineering Institute, Carnegie Mellon Univ., Pittsburgh, PA, USA, White Paper, Jul. 2018.
- [19] S. S. Joshi, P. Mukherjee, K. A. Davis, and J. C. Davis, "Introducing systems thinking as a framework for teaching and assessing threat modeling competency," in *Proc. ASEE Annu. Conf. & Expo.*, 2024, arXiv:2404.16632.
- [20] L. S. Vygotsky, *Mind in Society: The Development of Higher Psychological Processes*. Cambridge, MA, USA: Harvard Univ. Press, 1978.
- [21] M. Pedaste *et al.*, "Phases of inquiry-based learning: Definitions and the inquiry cycle," *Educ. Res. Rev.*, vol. 14, pp. 47–61, 2015.
- [22] A. Collins, J. S. Brown, and S. E. Newman, "Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics," in *Knowing, Learning, and Instruction: Essays in Honor of Robert Glaser*, L. B. Resnick, Ed. Hillsdale, NJ, USA: Lawrence Erlbaum, 1989, pp. 453–494.
- [23] D. Wood, J. S. Bruner, and G. Ross, "The role of tutoring in problem solving," *J. Child Psychol. Psychiatry*, vol. 17, no. 2, pp. 89–100, 1976.
- [24] D. A. Kolb, *Experiential Learning: Experience as the Source of Learning and Development*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1984.
- [25] M. L. Gick and K. J. Holyoak, "Schema induction and analogical transfer," *Cogn. Psychol.*, vol. 15, no. 1, pp. 1–38, 1983.
- [26] D. N. Perkins and G. Salomon, "Transfer of learning," in *International Encyclopedia of Education*, 2nd ed. Oxford, U.K.: Pergamon, 1992.
- [27] B. S. Bloom *et al.*, *Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain*. New York, NY, USA: David McKay, 1956.
- [28] L. W. Anderson and D. R. Krathwohl, Eds., *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. New York, NY, USA: Longman, 2001.
- [29] M. Nusrat, H. R. Mahi, S. Bhuiyan, and G. Hossain, "Mitigating the Information Cocoon Effect in Cognitively Aligned Recommendations: A Human-Centered Approach", in *Proc. IEEE 16th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, USA, 2026, pp. 324–330, doi: 10.1109/CCWC67433.2026.11393796.
- [30] T. Abdelghani, "Implementation of defense in depth strategy to secure industrial control system in critical infrastructures," *Amer. J. Artif. Intell.*, vol. 3, no. 2, pp. 17–24, 2019.
- [31] R. Ross, M. McEvelley, and J. Carrier Oren, "Systems security engineering: Considerations for a multidisciplinary approach in the engineering of trustworthy secure systems," NIST, Special Publication 800-160, Nov. 2016.
- [32] PERFECT Project, "Providing environment and resources for STEM 9th–12th graders in effective cybersecurity training," University of North Texas, 2024, [Online]. Available: <https://ci.unt.edu/perfect-project/index.html>.
- [33] M. Nusrat, T. Raheem, M. Shin, and G. Hossain, "Bridging Industry Standards with the PERFECT Project: A Foundational Cybersecurity Education in High Schools", *Preprint*, University of Bahrain, Jul. 16, 2026. [Online]. Available: <https://journal.uob.edu.bh/items/3db81670-3650-4cd9-b03d-2618ec328db4>
- [34] N. Mary and G. Hossain, "Towards Personalized Recommender System: A Gray-Box Modeling Approach," in *Proc. IEEE Int. Conf. Consumer Electronics (ICCE)*, Las Vegas, NV, USA, 2025, pp. 1–6.
- [35] Computer Science Teachers Association, "CSTA K–12 computer science standards, revised 2017," 2017, [Online]. Available: <https://www.csteachers.org/standards>.
- [36] R. Petersen, D. Santos, M. C. Smith, K. A. Wetzel, and G. Witte, "Workforce framework for cybersecurity (NICE framework)," NIST, Special Publication 800-181 Rev. 1, Nov. 2020.
- [37] T. Raheem, M. Nusrat, and G. Hossain, "A SWOT analysis of mobile privacy and security education for K–12 students," in *Proc. 14th Int. Symp. Digital Forensics and Security (ISDFS)*, Boston, MA, USA, 2026, pp. 1–6.
- [38] Joint Task Force Transformation Initiative, "Guide for conducting risk assessments," NIST, Gaithersburg, MD, USA, Special Publication 800-30 Rev. 1, 2012.
- [39] M. Tehranipoor and F. Koushanfar, "A survey of hardware Trojan taxonomy and detection," *IEEE Des. Test Comput.*, vol. 27, no. 1, pp. 10–25, Jan. 2010.
- [40] J. Rajendran, Y. Pino, O. Sinanoglu, and R. Karri, "Security analysis of logic obfuscation," in *Proc. 49th Annu. Design Automation Conf.*, San Francisco, CA, USA, Jun. 2012, pp. 83–89.
- [41] P. Subramanyan, S. Ray, and S. Malik, "Evaluating the security of logic encryption algorithms," in *Proc. IEEE Int. Symp. Hardware Oriented Security and Trust (HOST)*, Washington, DC, USA, May 2015, pp. 137–143.
- [42] X. Wang, D. Zhang, M. He, D. Su, and M. Tehranipoor, "Secure scan and test using obfuscation throughout supply chain," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 37, no. 9, pp. 1867–1880, Sep. 2018.
- [43] A. Cui, Y. Luo, and C.-H. Chang, "Static and dynamic obfuscations of scan data against scan-based side-channel attacks," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 2, pp. 363–376, Feb. 2017.
- [44] S. Khalaj Monfared, K. Mitard, A. Cannon, D. Forte, and S. Tajik, "LaserEscape: Detecting and mitigating optical probing attacks," in *Proc. 43rd IEEE/ACM Int. Conf. Computer-Aided Design (ICCAD)*, New York, NY, USA, Oct. 2024, pp. 1–10.
- [45] M. Yasin, B. Mazumdar, J. J. V. Rajendran, and O. Sinanoglu, "SAR-Lock: SAT attack resistant logic locking," in *Proc. IEEE Int. Symp. Hardware Oriented Security and Trust (HOST)*, McLean, VA, USA, May 2016, pp. 236–241.
- [46] Y. Xie and A. Srivastava, "Anti-SAT: Mitigating SAT attack on logic locking," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 38, no. 2, pp. 199–207, Feb. 2019.
- [47] M. Yasin, A. Sengupta, M. T. Nabeel, M. Ashraf, J. Rajendran, and O. Sinanoglu, "Provably-secure logic locking: From theory to practice," in *Proc. ACM SIGSAC Conf. Computer and Communications Security (CCS)*, Dallas, TX, USA, Oct. 2017, pp. 1601–1618.
- [48] A. Jonsson and G. Svingby, "The use of scoring rubrics: Reliability, validity and educational consequences," *Educ. Res. Rev.*, vol. 2, no. 2, pp. 130–144, 2007.
- [49] S. M. Brookhart, *How to Create and Use Rubrics for Formative Assessment and Grading*. Alexandria, VA, USA: ASCD, 2013.
- [50] M. Nusrat, G. Hossain, Kinshuk, and S. Bhuiyan, "Learning-to-explain through 20Q gaming: An explainable recommender for cybersecurity education," in *Proc. IEEE 16th Annu. Information Technology, Electronics and Mobile Communication Conf. (IEMCON)*, Berkeley, CA, USA, 2025, pp. 381–388.