Security (SEC)

Preamble

The world increasingly relies on computing infrastructure to support nearly every facet of modern critical infrastructure: transportation, communication, healthcare, education, energy generation and distribution, just to name a few. In recent years, with rampant attacks on and breaches of this critical computing infrastructure, it has become clearer that computer science graduates have an increased role in designing, implementing, and operating software systems that are secure and can keep information private.

In CS2023, the Security (SEC) Knowledge Area (KA) focuses on developing a *security mindset* into the overall ethos of computer science graduates so that security is inherent in all of their work products. The *Security* title choice was intentional to serve as a one-word umbrella term for this KA, which also includes concepts such as privacy, cryptography, secure system design, principles of modularity, and others that are imported from the other KAs. Reasons for this choice are discussed below; see also Figure 1.

The SEC KA also relies on shared concepts pervasive in all the other areas of CS2023. Additionally, the six cross-cutting themes of cybersecurity, as defined Cybersecurity Curricular 2017 (CSEC2017)[1], viewed with a computer science lens: confidentiality, integrity, availability, risk assessment, systems thinking, and adversarial thinking, are relevant here. In addition, the SEC KA adds a seventh cross-cutting theme: *human-centered thinking*, emphasizing that humans are also a link in the overall chain of security, a theme that needs to be inculcated into computer science students, along with *risk assessment* and *adversarial thinking,* which are not typically covered in other Computer Science Knowledge Areas (KAs). Students also need to learn security concepts such as authentication, authorization, and non-repudiation. They need to learn about system vulnerabilities and understand threats against software systems.

Principles of protecting systems (also in the Software Development Fundamentals and Software Engineering KAs) include security-by-design, privacy-by-design, and defense in depth. Another concept important in the SEC KA is the notion of assurance, which is an attestation that security mechanisms need to comply with the security policies that have been defined for data, processes, and systems. Assurance is tied in with the concepts verification and validation in the SE KA. With the increased use of computing systems and data sets in modern society, the issues of privacy, especially its technical aspects not covered in the Society, Ethics and Professionalism KA, become essential to computer science students.

Changes since CS 2013

[1] Joint Task Force on Cybersecurity Education. 2017. Cybersecurity Curricula 2017. ACM, IEEE-CS, AIS SIGSEC, and IFIP WG 11.8. https://doi.org/10.1145/3184594

The Security KA is an “updated” name for CS2013’s Information Assurance and Security (IAS) knowledge area. Since 2013, Information Assurance and Security has been rebranded as Cybersecurity, which has become a new computing discipline: the CSEC2017 curricular guidelines for this discipline have been developed by a Joint Task Force of the ACM, IEEE Computer Society, AIS and IFIP.

Moreover, since 2013, other curricular recommendations for cybersecurity beyond CS2013 and CSEC 2017 have been available. In the US, the Centers of Academic Excellence has Cyber Defense and Cyber Operations designations for institutions with cybersecurity programs that meet the CAE curriculum requirements, which are highly granular. Additionally, the US National Institute for Standards and Technologies (NIST) has developed and revised the National Initiative for Cybersecurity Education (NICE) Workforce Framework for Cybersecurity (NICE Framework), which identifies competencies (knowledge, and skills) needed to perform cybersecurity work. The European Cybersecurity Skills Framework (ECSF) includes a standard ontology to describe cybersecurity tasks, role and to address the cybersecurity shortage in EU member countries, and types. The computer science aspects of these guidelines also informed the content of this draft of the SEC KA.

Building on CS2013’s recognition of the pervasiveness of security in computer science, the CS2023 SEC KA focuses on ensuring that students develop the *security mindset* so that they are prepared for the continual changes occurring in computing. One noteworthy addition is the knowledge unit for security analysis and engineering to support the concepts of security-by-design and privacy-by-design.

Differences between the CS2023 Security KA and Cybersecurity

Feedback to earlier drafts of the SEC KA showed the need to clarify the differences between CS2023 SEC KA and the young computing-based discipline of cybersecurity. CS2023’s SEC KA, which is informed by the notion of a computer science disciplinary lens mentioned in CSEC 2017, focuses on those aspects of security, privacy, and related concepts important for computer science students. It builds primarily on security concepts already included in other CS2023 KAs. In short, the major goal of the SEC KA is to ensure computer science graduates to design and develop more secure code, ensure data security and privacy, and can apply the security mindset to their daily activities.

Protecting what happens *within* the perimeter is a core competency of computer science graduates. Although the computer science and cybersecurity knowledge units have overlaps, the demands upon cybersecurity graduates typically are to protect the perimeter. Cybersecurity is a highly interdisciplinary field of study that covers eight knowledge areas (data, software, component, connection, system, human, organizational, and societal security) and prepares its students for both technical and managerial positions. The first five knowledge areas are technical and have overlaps with the CS2023 SEC KA, but the intent of coverage is substantially different.

For instance, consider the data security knowledge unit. The computer science student will need to view this knowledge unit using the lens of computer science, as an extension of the material covered in CS2023's Data Management KA while the cybersecurity student will need to view data security in the overall context of cybersecurity goals. These viewpoints are not totally distinct and have overlaps, but the lenses used to examine and present the content are different, as shown in Figure 1. x1Similar diagrams apply to the CS2023 SEC KAs overlaps with the CSEC 2017 KAs.

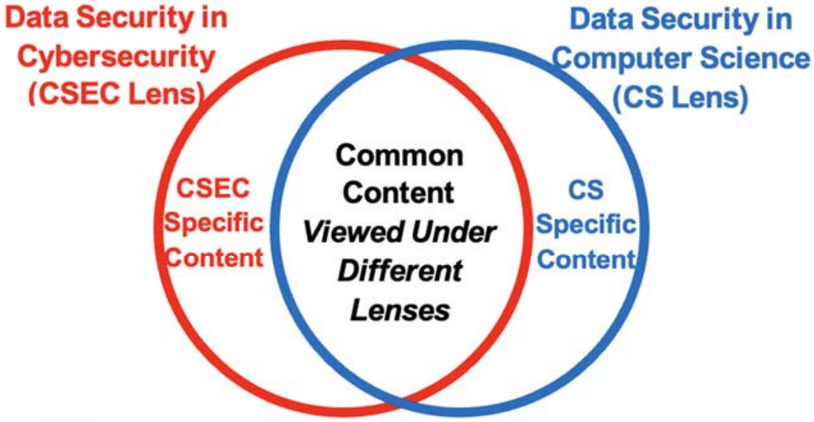
Data Security in Cybersecurity (CSEC Lens)
Data Security in Computer Science (CS Lens)
CSEC Specific Content
Common Content *Viewed Under Different Lenses*
CS Specific Content

Figure 1. Data Security – Cybersecurity versus CS2023 SEC. (Other knowledge areas will have similar Venn diagrams)

Core Hours

Knowledge Unit	**CS Core**	**KA Core**
Foundational Security	2	6
Defensive Programming	2	5
Cryptography	1	4
Security Analysis and Engineering	1	9
Digital Forensics	0	6
Security Governance	0	3
Total	**6**	**33**

The SEC KA also relies on CS Core and KA Core hours from the other KAs, as discussed below in the Shared Concepts and Crosscutting Themes section. At least 28 hours of CS Core hours from the other KAs are needed, either to provide the basis for the SEC KA or to complement its content shown here.

Knowledge Units

SEC-Foundations: Foundational Security

CS Core:

1. Developing a security mindset, including crosscutting concepts: confidentiality, integrity, availability, risk assessment, systems thinking, adversarial thinking, human-centered thinking
2. Vulnerabilities, threats, and attack vectors
3. Denial of Service (DoS) and Distributed Denial of Service (DDoS)
4. Principles and practices of protection, e.g., least privilege, open design, fail-safe defaults, and defense in depth; and how they can be implemented
5. Principles and practices of privacy
6. Authentication and authorization
7. Tensions between security, privacy, performance, and other design goals
8. Applicability of laws and regulations on security and privacy
9. Ethical considerations for designing secure systems and maintaining privacy

KA Core:

10. Cryptographic building blocks, e.g., symmetric encryption, asymmetric encryption, hashing, and message authentication
11. Hardware considerations in security
12. Access control, e.g., discretionary, mandatory, role-based, and attribute-based
13. Intrusion detection systems
14. Principles of usable security and human-centered computing
15. Concepts of trust and trustworthiness
16. Applications of security mindset: web, cloud, and mobile devices.
17. Internet of Things (IoT) security and privacy
18. Newer access control approaches

Illustrative Learning Outcomes:

CS Core:

1. Evaluate a system for possible attacks that can be launched by any adversary
2. Design and develop approaches to protect a system from a set of identified threats
3. Design and develop a system designed to protect individual privacy

KA Core:

4. Evaluate a system for trustworthiness

5. Develop a system that incorporates various principles of security
6. Design and develop a web application ensuring data security and privacy
7. Evaluate a system for compliance to a given law
8. Show a system has been designed to avoid harm to user privacy

SEC-Defense: Defensive Programming

CS Core Topics

1. Common vulnerabilities and weaknesses
2. Input validation and data sanitization
3. Type safety and type-safe languages
4. Buffer overflows, stack smashing, and integer overflows
5. SQL injection and other injection attacks
6. Security issues due to race conditions

KA Core Topics

7. Using third-party components securely
8. Assurance: testing (including fuzzing), verification and validation
9. Static and dynamic analyses
10. Preventing information flow attacks
11. Offensive security: what, why. where, and how
12. Malware: varieties, creation, and defense against them
13. Ransomware and its prevention

Non-core Topics (including Emerging topics)

14. Secure compilers and secure code generation

Illustrative Learning Outcomes

CS Core

1. Explain the problems underlying in provided examples of an enumeration of common weaknesses, and how they can be circumvented
2. Explain the importance of defensive programming in showing compliance to various laws
3. Apply input validation and data sanitization techniques to enhance security of a program
4. Rewrite a program in a type-safe language (e.g., Java or Rust) originally written in an unsafe programming language, (e.g., C/C++)
5. Evaluate a program for possible buffer overflow attacks and rewrite to prevent such attacks
6. Evaluate a set of related programs for possible race conditions and prevent an adversary from exploiting them
7. Evaluate and prevent SQL injections attacks on a database application.

KA Core

8. Explain the risks with misusing interfaces with third-party code and how to correctly use third-party code

9. Discuss the need to update software to fix security vulnerabilities and the lifecycle management of the fix
10. List examples of information flows and prevent unauthorized flows
11. Demonstrate how programs are tested for input handling errors
12. Use static and dynamic tools to identify programming faults
13. Describe different kinds of malicious software and how to prevent them from occurring in a system
14. Explain what ransomware is and implement preventive techniques to reduce its occurrence

SEC-Cryptography

CS Core Topics

1. Mathematical preliminaries: modular arithmetic, Euclidean algorithm, probabilistic independence, linear algebra basics, number theory, finite fields, complexity, asymptotic analysis
2. Differences between algorithmic, applied, and math views of cryptography
3. History and real-world applications, e.g., electronic cash, secure channels between clients and servers, secure electronic mail, entity authentication, device pairing, voting systems
4. Classical cryptosystems, such as shift, substitution, transposition ciphers, code books, machines.
5. Basic cryptography: symmetric key and public key cryptography
6. Kerckhoff's principle and use of vetted libraries

KA Core Topics

7. Additional mathematical foundations: primality and factoring; elliptic curve cryptography
8. Private-key cryptosystems: substitution-permutation networks, linear cryptanalysis, differential cryptanalysis, DES, AES
9. Public-key cryptosystems: Diffie-Hellman, RSA
10. Data integrity and authentication: hashing, digital signatures
11. Cryptographic protocols: challenge-response authentication, zero-knowledge protocols, commitment, oblivious transfer, secure 2-party or multi-party computation, secret sharing, and applications
12. Attacker capabilities: chosen-message attack (for signatures), birthday attacks, side channel attacks, fault injection attacks.
13. Quantum cryptography
14. Blockchain and cryptocurrencies

Illustrative Learning Outcomes

CS Core

1. Describe the role of cryptography in supporting security and privacy
2. Describe the dangers of inventing one's own cryptographic methods
3. Describe the role of cryptography in supporting confidentially and privacy

4. Discuss the importance of prime numbers in cryptography and explain their use in cryptographic algorithms
5. Implement and cryptanalyze classical ciphers

KA Core

6. Describe modern private-key cryptosystems and ways to cryptanalyze them
7. Describe modern public-key cryptosystems and ways to cryptanalyze them
8. Compare different algorithms in their support for security
9. Explain key exchange protocols and show approaches to reduce their failure
10. Describe real-world applications of cryptographic primitives and protocols
11. Describe quantum cryptography and the impact of quantum computing on cryptographic algorithms

SEC-Engineering: Security Analysis and Engineering

CS Core Topics

1. Security engineering goals: building systems that remain dependable despite errors, accidents, or malicious adversaries
2. Problem analysis and situational analysis to address system security
3. Security Design, including security testing; security evaluation and assessment
4. Tradeoff analysis based on time, cost, risk tolerance, risk acceptance, return on investment, and so on

KA Core Topics

5. Security Analysis, covering security requirements analysis; security controls analysis; threat analysis; and vulnerability analysis
6. Security Attack Domains and Attack Surfaces, e.g., communications and Networking, hardware, physical, social engineering, software, and supply chain
7. Security Attack Modes, Techniques and Tactics, e.g., authentication abuse; brute force; buffer manipulation; code injection; content insertion; denial of service; eavesdropping; function bypass; impersonation; integrity attack; interception; phishing; protocol analysis; privilege abuse; spoofing; and traffic injection
8. Security Technical Controls: identity and credential subsystems; access control and authorization subsystems; information protection subsystems; monitoring and audit subsystems; integrity management subsystems; cryptographic subsystems

Illustrative Learning Outcomes

CS Core

1. Create a threat model for a system or system design
2. Apply situational analysis to develop secure solutions under a specified scenario
3. Evaluate a give scenario for tradeoff analysis for system performance, risk assessment, and costs

KA Core

4. Design a set of technical security controls, countermeasures and information protections to meet the security requirements and security objectives for a system
5. Develop a system that incorporates various principles of security
6. Evaluate the effectiveness of security functions, technical controls and componentry for a system.
7. Identify security vulnerabilities and weaknesses in a system
8. Mitigate threats, vulnerabilities and weaknesses in a system

SEC-Forensics: Digital Forensics

CS Core Topics

1. Not applicable

KA Core Topics

2. Basic principles and methodologies for digital forensics.
3. System design for forensics
4. Rules of evidence – general concepts and differences between jurisdictions
5. Legal issues: digital evidence protection and management, chains of custody, reporting, serving as an expert witness
6. Forensics in different situations: operating systems, file systems, application forensics, web forensics, network forensics, mobile device forensics, use of database auditing
7. Attacks on forensics and preventing such attacks

Illustrative Learning Outcomes

CS Core

1. Not applicable

KA Core

2. Explain what a digital investigation is and how it can be implemented
3. Design and implement software to support forensics
4. Describe legal requirements for using seized data and its usage
5. Describe and implement end-to-end chain of custody from initial digital evidence seizure to evidence disposition
6. Extract data from a hard drive to comply with the law
7. Describe a person's professional responsibility and liability when testifying as a forensics expert
8. Recover data based on a given search term from an imaged system
9. Reconstruct data and events from an application history, or a web artifact, or a cloud database, or a mobile device
10. Capture and interpret network traffic
11. Discuss the challenges associated with mobile device forensics
12. Apply forensics tools to investigate security breaches
13. Identify and mitigate anti-forensic methods

SEC-Governance: Security Governance

CS Core Topics

1. Not applicable

KA Core Topics

2. Protecting critical assets from threats
3. Security governance: organizational objectives and general risk assessment
4. Security management: achieve and maintain appropriate levels of confidentiality, integrity, availability, accountability, authenticity, and reliability
5. Approaches to identifying and mitigating risks to computing infrastructure:
6. Security controls: management, operational and technical controls
7. Policies for data collection, backups, and retention; cloud storage and services; breach disclosure

Illustrative Learning Outcomes

CS Core

1. Not applicable

KA Core

2. Describe critical assets and how they can be protected
3. Differentiate between security governance, management, and controls, giving examples of each
4. Describe a technical control and implement it
5. Identify and assess risk of programs and database applications causing breaches
6. Design and implement appropriate backups, given a policy
7. Discuss a breach disclosure policy based on legal requirements and implement the policy
8. Identify the risks and benefits of outsourcing to the cloud.

CS Core Topics

1. TBD
2.

KA Core Topics

3. TBD

Illustrative Learning Outcomes

CS Core

14. TBD

KA Core
15. TBD

Expand the following as knowledge units?

SEC-Additional KUs or else?

Adversarial Machine Learning

CyberAnalytics

Hardware and Security

Professional Dispositions

- **Meticulous:** students need to pay careful attention to details to ensure the protection of real-world software systems.
- **Self-directed**: students must be ready to deal with the many novel and easily unforeseeable ways in which adversaries might launch attacks.
- **Collaborative**: students must be ready to collaborate with others , as collective knowledge and skills will be needed to prevent attacks, protect systems and data during attacks, and plan for the future after the immediate attack has been mitigated.
- **Responsible:** students need to show responsibility when designing, developing, deploying, and maintaining secure systems, as their enterprise and society is constantly at risk.
- **Accountable**: students need to know that as future professionals that they will be held accountable if a system or data breach were to occur, which should strengthen their resolve to prevent such breaches from occurring in the first place.

Math Requirements

Required:

- Sets, Relations, Logical Operations, Number Theory, Prime factoring
- Linear Algebra: Arithmetic Operations, Matrix operations
- Basic probability and descriptive statistics

Desired:

- System performance evaluation: probability and experiment design.
- Elliptic curves

Course Packaging Suggestions

There are two suggestions for course packaging.

The first is to infuse the CS Core hours of the SEC KA into appropriate places in other coursework that covers related security topics in the following knowledge units, as mentioned in the Shared Concepts section above. It seems to reasonable to assume that as the CS Core Hours of the SEC KA are only 6 hours, one or more of the following KUs being covered could accommodate the additional hours.

- AL-E: Algorithms and Society
- AR-D: Memory Hierarchy
- AR-H: Heterogeneous Architectures
- DM-I: Data Security and Privacy
- FPL-H: Language Translation and Execution
- FPL-N: /Runtime Behavior and Systems
- FPL-G: Type Systems
- HCI/Human Factors and Security
- NC-F: Network Security
- OS-G: Protection and Safety
- PDC-B: Communication
- PDC-D: Software Engineering
- SDF-A: Fundamental Programming Concepts and Practices
- SDF-D: Software Development Practices
- SE-F: Software Verification and Validation
- SEP-E: Privacy and Civil Liberties
- SEP-J: Security Policies, Laws and Computer Crime
- SF-G: Systems Security
- SPD-A: Common Aspects/Shared Concerns
- SPD-C: Mobile Platforms
- SPD-B: Web Platforms

The second approach is to create an additional course that packages the following:

Introduction to Computer Security to include the following:

- SEC-A: Foundational Security – 8 hours
- SEC-B: Defensive Programming – 7 hour
- SEC-C: Cryptography – 5 hours
- SEC-D: Security Analysis and Engineering – 4 hours
- SEC-E: Digital Forensics – 2 hours
- SEC-F: Security Governance – 1 hour
- AL-E: Algorithms and Society – 1 hour
- DM-I: Data Security and Privacy – 1 hour
- FPL-H: Language Translation and Execution – 1 hour

- FPL-N: Runtime Behavior and Systems – 1 hour
- FPL-G: Type Systems – 1 hour
- HCI/Human Factors and Security – 1 hour
- NC-F: Network Security – 3 hours
- OS-G: Protection and Safety – 3 hours
- PDC-B: Communication – 1 hour
- PDC-D: Software Engineering – 2 hours
- SDF-A: Fundamental Programming Concepts and Practices – 1 hour
- SDF-D: Software Development Practices – 1 hour
- SE-F: Software Verification and Validation – 2 hours
- SEP-E: Privacy and Civil Liberties – 1 hour
- SEP-J: ecurity Policies, Laws and Computer Crime – 2 hours
- SF-G: Systems Security – 2 hours
- SPD-A: Common Aspects/Shared Concerns – 2 hours
- SPD-C: Mobile Platforms – 2 hours
- SPD-B: Web Platforms – 2 hours

The coverage exceeds 45 lecture hours, and so in a typical 3-credit semester course, instructors would need to decide what topics to emphasize and what not to cover without losing the perspective that the course should help students develop the *security mindset*.

Prerequisites:

- Depends on the selected topics

Skill statement:

- A student who completes this course should develop the security mindset and be ready to apply this mindset to problems to securing software and systems

Committee

Chair: Rajendra K. Raj, Rochester Institute of Technology, Rochester, NY, USA

Members:

- Vijay Anand, University of Missouri – St. Louis, MO, USA
- Diana Burley, American University, Washington, DC, USA
- Sherif Hazem, Central Bank of Egypt, Egypt
- Michele Maasberg, United States Naval Academy, Annapolis, MD, USA
- Sumita Mishra, Rochester Institute of Technology, Rochester, NY, USA
- Nicolas Sklavos, University of Patras, Patras, Greece
- Blair Taylor, Towson University, MD, USA
- Jim Whitmore, Dickinson College, Carlisle, PA, USA

Contributors:

- Markus Geissler, Cosumnes River College, CA, USA
- Daniel Zappala, Brigham Young University, UT, USA