Information Assurance and Security in the ACM/IEEE CS2013

Ronald Dodge

Information Assurance and Security in the ACM/IEEE CS2013

Ronald C Dodge

United States Military Academy, West Point, NY, USA
ronald.dodge@usma.edu

Abstract. The ACM/IEEE Computing Curriculum 2013 is a community effort with representation from Academia and industry to outline curricular recommendations for undergraduate Computer Science degree programs. The effort began in 1968 [1] and conducts a complete review every ten years. The previous complete review was completed in 2001. [2] The current 2013 review is being developed to incorporate rapidly changing topics as technology and the world's use of technology evolves; however must do so within the curricular constraints of a complete undergraduate curriculum. The construction of the CS2013 is due for completion in December 2013. This effort describes the architecture of CS2013 and the details of the creation of a new knowledge area for Information Assurance and Security in the CS2013 computing curriculum.

1 Introduction

The world that our undergraduates must be prepared to succeed in is changing rapidly. Topics or study areas that seemed critical when a student began his or her undergraduate program may be out of date or encompassed by another emerging topic by the time they graduate. The Association for Computing Machinery and the Institute of Electrical and Electronics Engineers Computer Society (IEEE CS) has a long standing interest in providing input into the educational programs, teaching the future professionals in the field. Since 1968 [1], a joint commission from the two bodies has created a set of curricular recommendations for institutions to use in shaping the Computer Science undergraduate curriculum. This effort has traditionally been fully reviewed every 10 years with a minor interim assessment at the five year mark. The last major review was completed in 2001 and the interim review was completed in 2008. [2][3] Each new version of the ACM/ IEEE Computing Curriculum provides an opportunity for undergraduate Computer Science programs to review and assess their curriculum against a community constructed set of objectives. The process this year included many changes from previous years, including formally creating a knowledge area for Information Assurance and Security. This new knowledge area is unique among the slate of 18 knowledge areas due to its prevalence throughout all knowledge areas. In this paper, we begin in section two by providing further discussion of the CS2013 process and structure. In section three, we discuss in depth the Information Assurance and Security knowledge area. In section four, we conclude

adfa, p. 1, 2011.

with observations about the general state if institutions capacity to support the security goals of the CS2013 Body of Knowledge.

2 The Joint ACM/ IEEE Computer Science 2013

The ACM and IEEE-Computer Society chartered the CS2013 effort with the following directive:

> *To review the Joint ACM and IEEE-CS Computer Science volume of Computing Curricula 2001 and the accompanying interim review CS 2008, and develop a revised and enhanced version for the year 2013 that will match the latest developments in the discipline and have lasting impact.*
>
> *The CS2013 task force will seek input from a diverse audience with the goal of broadening participation in computer science. The report will seek to be international in scope and offer curricular and pedagogical guidance applicable to a wide range of institutions. The process of producing the final report will include multiple opportunities for public consultation and scrutiny*

The ACM and IEEE each appointed two co-chairs to manage the process and select the steering committee members. The group began work in the fall of 2010, beginning its work by reviewing the previous ACM/IEEE computing curriculum body of knowledge and preparing a survey to collect and validate existing topics and identify new and emerging requirements. The committee has met approximately every six months in person, supported with monthly teleconferences. The analysis resulted in the committee establishing the following goals:

1. Computer Science curricula should be designed to provide students with the flexibility to work across many disciplines.
2. Computer Science curricula should be designed to prepare graduates for a variety of professions, attracting the full range of talent to the field
3. CS2013 should provide guidance for the expected level of mastery of topics by graduates.
4. CS 2013 must provide realistic, adoptable recommendations that provide guidance and flexibility, allowing curricular designs that are innovative and track recent developments in the field.
5. The CS2013 guidelines must be relevant to a variety of institutions.
6. The size of the essential knowledge must be managed.
7. Computer Science curricula should be designed to prepare graduates to succeed in a rapidly changing field.
8. CS2013 should identify the fundamental skills and knowledge that all computer science graduates should possess while providing the greatest flexibility in selecting topics.

9. CS2013 should provide the greatest flexibility in organizing topics into courses and curricula.
10. The development and review of CS2013 must be broadly based.

The review of the prior bodies of knowledge and the feedback from a survey (described in paragraph 2.1) established the initial set of knowledge areas (KA). In this set of 18 KA's, six new areas emerged. Of particular note is the inclusion of the Information Assurance and Security Knowledge Area (IAS). Each KA was assigned a chair and at least two other committee members.

- AL-Algorithms and Complexity
- AR-Architecture and Organization
- CN-Computational Science
- DS-Discrete Structures
- GV-Graphics and Visual Computing
- HCI-Human-Computer Interaction
- IAS-Information Assurance and Security (new in 2013)
- IM-Information Management
- IS-Intelligent Systems
- NC-Networking and Communication (new in 2013)
- OS-Operating Systems
- PBD-Platform-based Development (new in 2013)
- PD-Parallel and Distributed Computing (new in 2013)
- PL-Programming Languages
- SDF-Software Development Fundamentals (new in 2013)
- SE-Software Engineering
- SF-Systems Fundamentals (new in 2013)
- SP-Social Issues and Professional Practice

As the subcommittees produced drafts of their Knowledge Areas, others in the community were asked to provide feedback, both through presentations at conferences and direct review requests. The Steering Committee also collected community input through an online review and comment process. The KA subcommittee Chairs (as members of the CS2013 Steering Committee) worked to resolve conflicts, eliminate redundancies and appropriately categorize and cross-reference topics between the various KAs. Thus, the computer science community beyond the Steering Committee played a significant role in shaping the Body of Knowledge throughout the development of CS2013. This two-year process ultimately converged on the version of the Body of Knowledge presented in the IronMan draft. [4]

2.1 Survey Input

The development work of creating the initial set of KA's to include in the overall body of knowledge relied heavily on two measures of input; prior ACM/IEEE computer science curriculum work and a survey distributed to institutions worldwide. The

purpose of the survey was to gather information on area of knowledge that had either increased or decreased in importance. The survey was sent to over 3500 institutions (1500 in the United States and 2000 internationally), addressed primarily to Computer Science (and related discipline) department chairs and directors of undergraduate studies. The response rate was unfortunately lower than desired (201 responses), however produced valuable input. The respondents represented a wide variety of institution type and size:

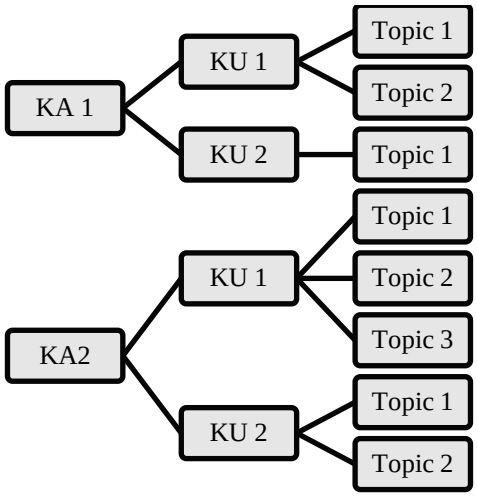
- research-oriented universities (55%)
- teaching-oriented universities (17.5%)
- undergraduate-only colleges (22.5%)
- community colleges (5%)

- less than 1,000 students (6.5%) 77
- 1,000 to 5,000 students (30%) 78
- 5,000 to 10,000 students (19%) 79
- more than 10,000 students (44.5%)

2.2 Overview of the CS2013 Body of Knowledge Structure

There are two fundamental concepts for the Body of Knowledge. The first and most fundamental is that the Knowledge Areas *are not* intended to correlate directly to a specific course. The intent is not to imply any restriction on how an institution may desire to address any of the content. The second concept is the tiered approach to describing the importance of given content. The Body of Knowledge is structure hierarchically as shown in Figure 1

Fig. 1 Body of Knowledge Structure



Each KA (as listed in paragraph 2), is broken down into sub areas called Knowledge Units. Each Knowledge Unit is described by a collection of topics and related learning outcomes. The learning outcomes are also represented by three degrees of mastery:

- ***Familiarity***: The student understands what a concept is or what it means.
- ***Usage***: The student is able to use or apply a concept in a concrete way.
- ***Assessment***: The student is able to consider a concept from multiple viewpoints and/or justify the selection of a particular approach to solve a problem.

As described earlier, it is expected that topics will span multiple courses. The topics are identified as either core-tier 1, core-tier 2, or elective. The core-tier 1 and core-tier 2 topics are further described by the number of hours that is expected within the undergraduate computer science curriculum. Each hour is reflective of the lecture or supervised learning hours within which the topic is one of the key learning objectives. These hours along with the learning outcomes are intended to provide guidance on the depth of coverage. Understanding that not all programs would or should be alike, the following guidance is provided:

- A curriculum should include all topics in the Tier-1 core and ensure that all students cover this material.
- A curriculum should include all or almost all topics in the Tier-2 core and ensure that all students cover the vast majority of this material.
- A curriculum should include significant elective material: Covering only "Core" topics is insufficient for a complete curriculum.

A much more detailed discussion of the motivation and philosophy behind the body of knowledge may be found in [4].

3 CS2013 Information Assurance and Security Knowledge Area

The Information Assurance and Security Knowledge Area is new in the CS2013 Body of Knowledge. It was clear both in the analysis from the steering committee and the survey results, that the broad range of topics defined by information security are an essential component of any undergraduate computer science program. This new KA was proposed as part of the very first steering committee meeting. During the World Conference on Information Security Education (WISE 7) in June, 2011, organized by an internationally focused IFIP technical working group (WG 11.8), the topic was discussed and further refined to use the title "Information Assurance and Security".

Information assurance has been defined as "a set of controls (technical and policy) intended to protect and defend information and information systems by ensuring their

availability, integrity, authentication, confidentiality, and non-repudiation. This includes providing for restoration of information systems by incorporating protection, detection, and reaction capabilities."[5] [6] This concept of assurance also carries a connotation of an attestation that past processes or data is valid.

Information assurance and security education, then, includes all efforts to prepare a workforce with the needed knowledge, skills, and abilities to assure our information systems and attest to the validity of the current state of processes and data. Information assurance and security education has been growing in importance and activity for the past two decades.

The McCumber model [7] [8] has been widely used over the past 10 years to broadly define the relationships between information states, security services, and security operations (called counter measures in the original model). As the security and assurance landscape has matured, the relationships should be clarified to address that the full spectrum operational aspect of security is not correctly contained by the term counter measures and that emergence of assurance of processes (current and past) is playing a critical role in the field of IAS.

3.1 The construction of the IAS KA

The aim of the IAS KA is to define the core (core-tier1 and core-tier2) and elective knowledge threads that depict what a computer science undergraduate should possess upon graduation.

The IAS KA is unique in the collection of KA's due to its pervasive nature in all KA's. One can express this cross cutting impact by comparing security to performance. In the past, many concepts in Computer Science were performance based. Algorithms were developed to increase the performance of memory utilization or database searches. In this light, the way we do things in Computer Science must be done securely. This is not to say that Information Assurance and Security does not have concepts that belong solely to the IAS KA. As shown in table 1, concepts that are unique to IAS are listed with specific core tier 1 and tier 2 hours.

Table 1. Information Assurance and Security Knowledge Units/Hours

Knowledge Unit	Core-Tier1	Core-Tier2	Electives
IAS/Foundational Concepts in Security	1	-	N
IAS/Principles of Secure Design	1	1	N
IAS/Defensive Programming	1	1	Y
IAS/Threats and Attacks	-	1	N
IAS/Network Security	-	2	Y
IAS/Cryptography	-	1	N
IAS/Web Security	-	-	Y
IAS/Platform Security	-	-	Y
IAS/Security Policy and Governance	-	-	Y
IAS / Digital Forensics	-	-	Y
IAS/Secure Software Engineering	-	-	Y

Each knowledge unit contains a collection of topics. For the knowledge unit, "IAS/Principles of Secure Design", the topics are shown below. Each topic has an associated learning outcome with a desired level of comprehension (Familiarity, Usage, and Assessment). While we also list the topics associated with the other KU's with the IAS KA, further detail (learning outcomes) and elective topics are left for review in [4]. As detailed below in the Principles of Security Design, the topics document cross reference locations where the referenced topic is also presented in another KA. As an example, the cross reference for the very first topic, "least privilege and isolation" extends to the KA's for Operating Systems, System Fundamentals, and Programming languages. Those references are listed after the Principles of Security Design.

IAS/Principles of Secure Design [1 Core-Tier1 hours, 1 Core-Tier2 hours]
Topics:
[Core-Tier1]

- Least privilege and isolation (cross-reference OS/Security and Protection/Policy/mechanism separation and SF/Virtualization and Isolation/Rationale for protection and predictable performance and PL/Language Translation and Execution/Memory management)
- Fail-safe defaults (cross-reference SE/Software Construction/ Coding practices: techniques, idioms/patterns, mechanisms for building quality programs and SDF/Development Methods/Programming correctness)
- Open design (cross-reference SE/Software Evolution/ Software development in the context of large, pre-existing code bases)
- End-to-end security (cross reference SF/Reliability through Redundancy/ How errors in-crease the longer the distance between the communicating entities; the end-to-end principle)
- Defense in depth
- Security by design (cross reference SE/Software Design/System design principles)
- Tensions between security and other design goals

[Core-Tier 2]

- Complete mediation
- Use of vetted security components
- Economy of mechanism (reducing trusted computing base, minimize attack surface) (cross reference SE/Software Design/System design principles and SE/Software Construction/Development context: "green field" vs. existing code base)
- Usable security (cross reference HCI/Foundations/Cognitive models that inform interaction design)
- Security composability
- Prevention, detection, and deterrence (cross reference SF/Reliability through Redundancy/Distinction between bugs and faults and NC/Reliable Data Delivery/Error control and NC/Reliable Data Delivery/Flow control)

Learning outcomes:
[Core-Tier1]

1. Describe the principle of least privilege and isolation and apply to system design [application]
2. Understand the principle of fail-safe and deny-by-default [familiarity]
3. Understand not to rely on the secrecy of design for security (but also that open design alone does not imply security) [familiarity]
4. Understand the goals of end-to-end data security [familiarity]
5. Understand the benefits of having multiple layers of defenses [familiarity]
6. Understand that security has to be a consideration from the point of initial design and throughout the lifecycle of a product [familiarity]
7. Understanding that security imposes costs and tradeoffs [familiarity]

[Core-Tier2]

8. Describe the concept of mediation and the principle of complete mediation [application]
9. Know to use standard components for security operations, instead of re-inventing fundamentals operations [familiarity]
10. Understand the concept of trusted computing including trusted computing base and attack surface and the principle of minimizing trusted computing base [application]
11. Understand the importance of usability in security mechanism design [familiarity]
12. Understand that security does not compose by default; security issues can arise at boundaries between multiple components [familiarity]
13. Understand the different roles of prevention mechanisms and detection/deterrence mechanisms [familiarity]

The cross reference topics that are documented in the first Principles of Security Design topic are:

OS/Security and Protection [2 Core-Tier2 hours]
Topics:

- Overview of system security
- Policy/mechanism separation
- Security methods and devices
- Protection, access control, and authentication
- Backups

SF/Virtualization and Isolation [2 Core-Tier 2 hours]
Topics:

- Rationale for protection and predictable performance
- Levels of indirection, illustrated by virtual memory for managing physical memory resources
- Methods for implementing virtual memory and virtual machines

PL/Language Translation and Execution [3 Core-Tier2 hours]
Topics (only includes 50% of listed topics)
- Run-time layout of memory: call-stack, heap, static data
  - Implementing loops, recursion, and tail calls
- Memory management
  - Manual memory management: allocating, de-allocating, and reusing heap memory
  - Automated memory management: garbage collection as an automated technique using the notion of reachability

3.2 Distributed nature of the IAS KA topics

The IAS KA is the most heavily cross referenced KA in the CS2013 Body of Knowledge. As can be inferred from the example provided in the preceding paragraph, the relatively small number of recommended curriculum hours is not representative of the presence of IAS topics and concepts in the CS2013 Body of Knowledge. As can be seen in Table 2, while IAS has 9 combined core hours, there are 63.5 hours distributed through the other KA's

Table 2: IAS Cross KA Hour Distribution

KA's	Core-Tier1	Core-Tier2	Elective
IAS	3	6	Y
IAS distributed in other KA's	32	31.5	Y

As an example of security topics that are addressed in KA's outside of IAS, System Development Fundamentals contains 10 Core-tier 1 hours that address important security concepts.

SDF/Development Methods [10 Core-Tier1 hours]
Topics:
- Program comprehension
- Program correctness
  - Types or errors (syntax, logic, run-time)
  - The concept of a specification
  - Defensive programming (e.g. secure coding, exception handling)
  - Code reviews
  - Testing fundamentals and test-case generation
  - Test-driven development
  - The role and the use of contracts, including pre- and post-conditions
  - Unit testing
- Simple refactoring
- Modern programming environments
  - Code search
  - Programming using library components and their APIs

- Debugging strategies
- Documentation and program style

4 Conclusions and Recommendations

The importance of security concepts and topics has emerged as a core requirement in the Computer Science discipline, much like the importance of performance concepts has been for many years. The development of the IronMan draft for the CS2013 computing curriculum has highlighted the emphasis programs are now placing on security topics. This development however has also identified some weaknesses in the capacity institutions have to adequately address the integration of the security. The emerging nature of security is reflected in the challenge of Computer Science programs and faculty with security experience to inculcate the security concepts in the breadth of courses.

5 References

1. ACM Curriculum Committee on Computer Science. 1968. Curriculum 68: 213 Recommendations for Academic Programs in Computer Science. *Comm. ACM* 11, 3 214 (Mar. 1968), 151-197.
2. ACM/IEEE-CS Joint Task Force on Computing Curricula. 2001. ACM/IEEE Computing Curricula 2001 Final Report. http://www.acm.org/sigcse/cc2001. 217
3. ACM/IEEE-CS Joint Interim Review Task Force. 2008. Computer Science Curriculum 221 2008: An Interim Revision of CS 2001, Report from the Interim Review Task Force. 222 http://www.acm.org/education/curricula/ComputerScience2008.pdf
4. ACM/IEEE-CS CS2013 IronMan draft, http://ai.stanford.edu/users/sahami/CS2013/ironman-draft/cs2013-ironman-v1.0.pdf
5. National Security Agency, http://www.nsa.gov/ia/iaFAQ.cfm?MenuID=10#1
6. NIST publication 800-53. http://csrc.nist.gov/publications/nistpubs/800-53-Rev2/sp800-53-rev2-final.pdf
7. Machonachy, W. Victor; Schou, Corey D.; Ragsdale, Daniel; Welch , Don; “A model for Information Assurance: An Integrated Approach”, Proceedings of the 2001 IEEE Workshop on Information Assurance and Security, United States Military Academy, West Point , NY, 5-6 June 2001
8. McCumber, John. “Information Systems Security: A Comprehensive Model”. Proceedings 14th National Computer Security Conference. National Institute of Standards and Technology. Baltimore, MD. October 1991