

Course Description

Department and Course Number:	CSc 236
Course Coordinator:	Cui Zhang, Professor of Computer Science
Course Title:	Formal Methods in Secure Software Engineering
Total Credits:	3 units

Current Catalog Description:

Basic concepts of formal methods in secure software engineering; formal methods based software development models and methodologies; formal methods for software system specification, modeling, analysis, verification, information assurance and security; systems and tools for the application of formal methods in secure software engineering; advances of formal methods in secure software engineering. **Prerequisite:** Fully classified graduate status in Computer Science or Software Engineering.

Textbook:

A. Harry, Formal Methods Fact File VDM and Z¹, John Wiley and Sons, 1996.

References:

1. Journal/conference papers and Ph.D. thesis works in the area of formal methods in secure software engineering.
2. J. Barnes, High Integrity Software: The Spark Approach to Safety and Security, Addison Wesley, 2002.
3. M. Croxford and R. Chapman, Praxis High Integrity Systems, “Correctness by Construction – A Manifesto for High Integrity Engineering”, Tutorial, IEEE International Symposium on Secure Software Engineering, Arlington, Virginia, March 13-15, 2006.
4. Formal Methods Specification and Analysis Guidebook for the Verification of Software and Computer Systems (Volume II: A Practitioner's Companion), NASA, (http://eis.jp1.nasa.gov/quality/Formal_Methods/).
5. R. Lai and A. Jirachiefoattana, Communication Protocol Specification and Verification, Kluwer Academic Publishers, 1998.
6. J. M. Spivey, The Z Notation: A Reference Manual, Prentice Hall, 1992 (<http://spivey.oriel.ox.ac.uk/~mike/zrm/>).
7. M. Kaufmann, P. Manolios, and J.S. Moore, Computer-Aided Reasoning, Kluwer Academic Publishers, 2000 (<http://www.cs.utexas.edu/users/moore/acl2>).

Course Goals:

1. To develop an appreciation of the strengths of formal methods for engineering secure software systems and to build up a solid background for the application of formal methods to various tasks of the software development process.

¹ This book has been chosen for its coverage of multiple formal methods including VDM (Vienna Development Method), Z (the Z Notation), RAISE (Rigorous Approach to Industrial Software Engineering), LOTOS (Language of Temporal Ordering Specification), and Estelle (Extended State Transition Language) at various depths.

2. To gain a basic level of competence in using formal methods to model software systems and to specify, analyze, and verify software system properties including security properties.

Prerequisites by Topic:

Fundamental concepts in the following areas:

1. Software engineering.
2. Operating systems.
3. High-level programming languages.
4. Computer networks.
5. Discrete mathematics and state transition models.

Major Topics Covered in the Course:

1. Introduction and mathematics background (3 hours).
2. Formal methods-based life-cycle models (1.5 hours).
3. Correctness by construction (CbyC) and the Spark approach² for secure software development (4.5 hours).
4. Requirements models, formal specification styles (3 hours).
5. Static analysis and verification, dynamic analysis and verification (3 hours).
6. Software development in VDM and object-orientation VDM (4.5 hours).
7. Software development in Z and object-oriented Z (4.5 hours).
8. Formal methods for information assurance and security; e.g., formal specification of security policies, formal specification-based intrusion detection (9 hours).
9. Formal methods for communication protocols, distributed systems and real-time systems (3 hours).
10. Systems and tools for the application of formal methods (e.g., ACL2³) (3 hours).
11. Advances of formal methods in secure software engineering (6 hours).

Laboratory Projects:

Homework assignments may include but not be limited to the following:

1. To model and specify, using the Z notation, an operating system kernel with both the access control model and security policies, e.g., the policy of Multiple Levels of Security (MLS).
2. To model and specify, using the Z notation, a realistic data centered software system, e.g., the library system, with features of concurrent processing, e.g., synchronization and mutual exclusion.

² CbyC and the Spark Approach have been practiced by National Security Agency for its software development.

³ ACL2 (the 2nd version of A Computational Logic) is a first-order logic-based automated theorem-proving system. It has been used in both hardware verification and software verification. Recently it has been used in the area of information assurance and security. ACL2 has been installed on one ECS server.

3. To use an automated theorem-proving system, e.g., ACL2, for formal specification and verification.

Oral and Written Communication:

One assignment on advances of formal methods for information assurance and security (e.g., formal specification-based intrusion detection, verification of security protocols) involves oral and written communication. Students are required to select topics from a list of topics prepared by the instructor. Under the guidance of the instructor, students study the topics, prepare notes and viewgraphs based on their study results, and give a 20-minute individual or team oral presentation to the class.

Social and Ethical Issue:

No significant component.

Theoretical Content:

Formal specification and verification.

Analysis and Design:

Formal methods-based static and dynamic analysis of software systems.

CZ

1/28/08