SE547: Lecture 1

Alan Jeffrey

Overview

Course summary

Timetable

Administrivia

Course overview

Homework

Course summary

This course provides an overview of foundational techniques in the specification and verification of computer systems in the presence of malicious attackers.

Topics may include: formal models of interaction, attacker models, robust safety properties such as confidentiality and authenticity, information flow properties such as noninterference, and tools such as model checkers, type checkers and theorem provers.

Timetable (approximate!)

Part I: Secure Protocols

1. *Overview*: getting our bearings.

2. *Foundational Calculi*: lambda-calculus and pi-calculus.

3. *A Calculus for Cryptographic Protocols*: the spi-calculus.

4. *Secrecy and Authenticity*: verifying properties of protocols.

5. *Logical Techniques*: BAN logic.

6. *Midterm exam*.

Timetable (approximate!)

Part II: Secure Software

7. *Implementing Logical Satisfaction*: Binary Decision Diagrams.

8. *Hardware Model Checking*: Using BDDs to verify hardware.

9. *Software Model Checking*: Using BDDs to verify software.

10. *Proof-Carrying Code*: Software verification as an alternative to digitally signed binaries.

11. *Final exam*

Administrivia: contact details

Lecturer: Alan Jeffrey

Email: ajeffrey@cs.depaul.edu

Office: CST 840

Phone: (312) 362 8322

Office hours: 3.30-5.00pm Tuesdays and Thursdays.

Administrivia: reading materials

Course home page: http://fpl.cs.depaul.edu/ajeffrey/se547/, contains lectures, homeworks, papers, pointers to tools...

No textbook: this course is based on research papers and on-line materials, not a textbook.

Administrivia: prerequisites

CSC 390: Fundamentals of Information Assurance

CSC 416 Foundations of Computer Science II

Administrivia: assessment

Midterm exam (12 Feb 2004): 25%

Final exam (18 Mar 2004): 25%

Weekly homeworks (submitted using Courses OnLine, best 7 out of 8): 50%

All students must attend the mid-term and final exams

Late assignments will not be accepted without medical evidence.

Plagiarism or collusion is unacceptable, and will earn an F in the course.

Week 1's reading

- C. Meadows. Formal Methods for Cryptographic Protocol Analysis: Emerging Issues and Trends. In *IEEE Journal on Selected Areas in Communication*, 21(1), pp. 44-54, 2003.
- J. Clark and J. Jacob. A Survey of Authentication Protocol Literature. Unpublished manuscript, 1997.

Foundations of Cryptographic Protocols

What is a communications protocol? A cryptographic algorithm? A cryptographic protocol?

What is 'perfect black box cryptography' (aka the Dolev-Yao model)?

What are typical goals of a cryptographic protocol?

Why is verifying cryptographic protocols often harder than verifying regular protocols?

Verifying properties of cryptographic protocols is undecidable: what does this mean? What can we do about it?

Formal Methods

What are formal methods? When are they appropriate?

Why might formal methods be appropriate for cryptographic protocols?

Sample formal methods: logics, state space exploration, and type systems. What are these?

Tools to support formal methods: theorem provers, model checkers, and type checkers. What are these?

Trends in Protocols

Why isn't this a done deal (aren't there only 25 cryptographic protocols)?

Symmetric key cryptography ruled the roost in the 1970s and 1980s; is that still true today?

Threat model in the 1970s and 1980s was compromise of read or write access to secrets; is that still true today?

Protocol analysis in the 1980s and 1990s was typically carried out after standardization was completed; is this a good idea?

Cryptographic Prerequisites

What is a cryptographic algorithm?

Symmetric key crypto? Block cipher? Stream cipher?
(Notation $\{ M \}K$)

Asymmetric key crypto? Why use symmetric rather than asymmetric crypto? (Notation $\{| M |\}K$)

Hash function? Why are these useful? (Notation $\#M$)

Protocol types

Typically M is a message, Kab is a symmetric key shared between honest agents A and B, S is a trusted third party, Ka is A's public key, and Ka^{-1} is A's private key.

Symmetric key without trusted third party, e.g. niave encryption:

(1) A -> B: { M }Kab

What are the goals of this protocol? Does it achieve them?

Protocol types

Typically M is a message, Kab is a symmetric key shared between honest agents A and B, S is a trusted third party, Ka is A's public key, and Ka^{-1} is A's private key.

Symmetric key with trusted third party, e.g. a flawed variant of Denning-Sacco:

(1) A -> S: A, B
(2) S -> A { Kab, { Kab }Kbs }Kas
(3) A -> B { Kab }Kbs, { M }Kab

What are the goals of this protocol? Does it achieve them?

Protocol types

Typically M is a message, Kab is a symmetric key shared between honest agents A and B, S is a trusted third party, Ka is A's public key, and Ka^{-1} is A's private key.

Asymmetric key, e.g. digital signature:

(1) A -> B: $CertA$, M, {| #M |}Ka^{-1}

What is $CertA$? What are the goals of this protocol? Does it achieve them?

Attacker model

Attacker goal: violate the security goals of the protocol.

Attacker capabilities: duplicate, edit, and spoof messages.

‘Man in the middle’ model: the network is the attacker.

What can’t the attacker do?

How can we model this?

Are there attack vectors we’re not modelling?

Next week

Homework sheet 1.

Fundamental calculi: lambda- and pi-calculus.