COM3005J: Agile Processes
Course Information

Dr. Anca Jurcut
E-mail: anca.jurcut@ucd.ie

School of Computer Science and Informatics
University College Dublin

Beijing-Dublin International College

About me

- I was born in Romania and I moved to Ireland in 2007, after awarded with a PhD Scholarship.

European Countries
How many European countries can you correctly identify on the map?
Colour the map and check your answers with an atlas.

- I am an Assistant Professor in School of Computer Science and Informatics in UCD as part of the BDIC program
- Before, I taught and I did research in the University of Limerick and I worked for IBM Dublin, Ireland
- Last Autumn semester it was my fourth time being in China!

About Me

- I did my PhD in University of Limerick (Ireland), in the area of Data and Network Security.
  - I do research in Security Protocols Analysis, Logic-based Verification Techniques, Cryptography, Automation of Logics used in Formal Verification, Theorem Proving Techniques and Mathematical Modelling

- Previously taught in:
  - West University of Timisoara, Romania
  - University of Limerick, Ireland
  - University College Dublin, Ireland
  - Oulu University, Finland

November 2013 Volume 8 Number 11
ISSN 1796-2021
JOURNAL OF COMMUNICATION
www.jocm.us
ETP

Certificate of Completion

SCIENCE AND ENGINEERING INSTITUTE

THIS CERTIFICATE IS PROUDLY AWARDED TO

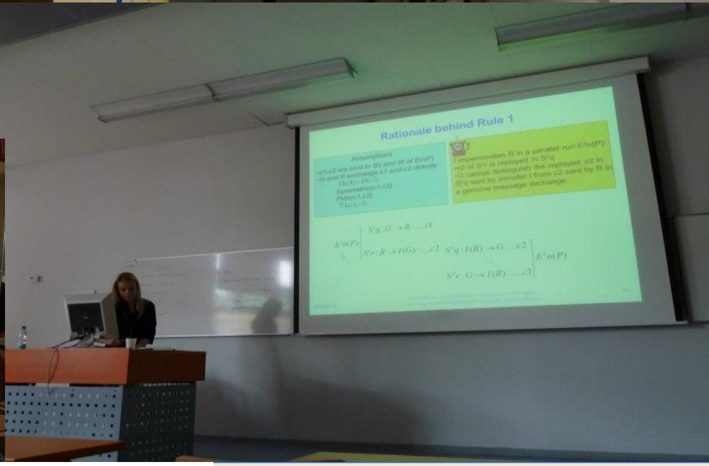
Anca Jurcut

Paper Title: Establishing and Fixing Security Protocols Weaknesses using a Logic-based Verification Tool(SX006)

for your excellent oral presentation at the conference and your significant contribution to the success of 2013 the 3rd International Conference on Communication and Network Security (ICCNS 2013) London, UK

Date

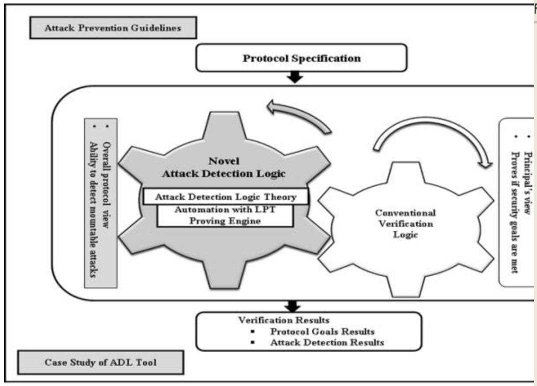
Dr. Alice Wu



Rationale behind Rule 1

Assumptions

I impersonates R in a parallel run E²n(P):
- c2 of S¹r is replayed in S²q
- G cannot distinguish the replayed c2 in S²q sent by intruder I from c2 sent by R in a genuine message exchange



Attack Prevention Guidelines

Protocol Specification

- Overall protocol view
- Ability to detect mountable attacks

Novel Attack Detection Logic

Attack Detection Logic Theory

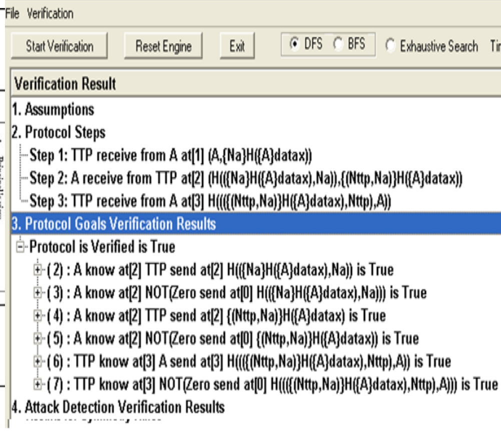
Automation with LPT Proving Engine

Conventional Verification Logic

- Principal's view
- Proves if security goals are met

Verification Results
- Protocol Goals Results
- Attack Detection Results

Case Study of ADL Tool



```
Verification Result
1. Assumptions
2. Protocol Steps
  Step 1: TTP receive from A at[1] (A,{Na}H({A}datax))
  Step 2: A receive from TTP at[2] (H(({Na}H({A}datax),Na)),{(Nttp,Na)}H({A}datax))
  Step 3: TTP receive from A at[3] H((({(Nttp,Na)}H({A}datax),Nttp),A))
3. Protocol Goals Verification Results
  ⊟ Protocol is Verified is True
    ⊞ ( 2) : A know at[2] TTP send at[2] H(({Na}H({A}datax),Na)) is True
    ⊞ ( 3) : A know at[2] NOT(Zero send at[0] H(({Na}H({A}datax),Na))) is True
    ⊞ ( 4) : A know at[2] TTP send at[2] {(Nttp,Na)}H({A}datax) is True
    ⊞ ( 5) : A know at[2] NOT(Zero send at[0] {(Nttp,Na)}H({A}datax)) is True
    ⊞ ( 6) : TTP know at[3] A send at[3] H((({(Nttp,Na)}H({A}datax),Nttp),A)) is True
    ⊞ ( 7) : TTP know at[3] NOT(Zero send at[0] H((({(Nttp,Na)}H({A}datax),Nttp),A))) is True
4. Attack Detection Verification Results
```

Course Description

- ✓ Agile processes has taken the software development industry by storm.
- ✓ Everyone wants to be agile, but what does it really mean and how do you achieve agile development?
- ✓ This computer science course cuts beyond the agile methodology hype and teaches you the fundamental agile concepts that span a wide range of methodologies.
- ✓ It analyses the key agile ideas, their benefits, their limitations, and how best to take advantage of them to enhance your software skills and show employers that you have mastered an essential component of today's IT industry.

Course Overview (1)

- Course is divided into 6 parts:

1) The **Agile manifesto** and the **context** of agile processes
2) Agile principles: what key methodological ideas underlie the agile movement?
3) Agile roles: how does agile redefine traditional software jobs and tasks, in particular the manager's role?
4) Agile practices: what are the concrete techniques that agile teams use to apply these processes?
5) Agile artifacts: what practical tools are essential to the work of agile developers?
6) Agile assessment: among agile ideas, which ones are essentially hyped and useless, which ones are actually harmful, and which ones will truly help you effectively produce high-quality software?

Course Overview (2)

- Course Schedule & Duration:
  - 12 weeks of lecturing & assessments (room 526 Teaching Building 3)
    - Each Odd Thursday (9:55am) - Lecture
    - Each Even Thursday (9:55am) – Assignment Work
    - **Now 11 weeks after today!**
    - **Next Thursday: Assignment Work** (Requirement: careful read of lecture slides)
  - One week revision time (Week 12)
  - Exam…

Course Overview (3)

- Benefits of this Course:
  - It takes a strictly objective view of agile processes, enabling you to retain the best agile principles and practices.
  - By analysing agile processes in depth and showing you how to benefit from them, it will make you a better developer, equipped to deal with the challenges of ambitious software projects.

Course Overview (4)

- What will you learn?
  - The Agile manifesto and the context of agile processes
  - Agile principles
  - Agile roles
  - Agile practices
  - Agile artifacts
  - Agile assessment

Course Overview (5)

- Course Structure
  - **Chapter 1: Context**
    - Introduction to the agile context: manifesto, principles, methods, and values.
  - **Chapter 2: Principles**
    - Detailed discussion on the agile principles.
  - **Chapter 3: Roles**
    - Detailed discussion on the roles in an agile project, and how they evolved from the traditional roles.
  - **Chapter 4: Practices**
    - Presentation of the main agile practices, namely meetings, development, releases, testing, management.
  - **Chapter 5: Artifacts**
    - Description of the main agile artifacts.
  - **Chapter 6: Assessment**
    - Final assessment on the agile methods.
  - **Final Exam**

Course Grading

- **Examination** **60%**
  - The sum of the points achieved in the final exam is worth 60% of the final grade
- **Assignments ("Quick questions")** **10%**
  - The sum of the points achieved in the Assigments ("Quick questions") altogether is worth 10% of the final grade. Each correct answer to a quick question is generally worth 1 point
- **“Homework quizzes” in class participation** **30%**
  - The sum of the points achieved in the Even Weeks Lectures time ("Homework" quizzes) altogether is worth 30% of the final grade. A correct answer to each quiz question is generally worth 1 point. Occasionally, there are questions that are made of several sub-questions, each worth one point. One quiz is droppable (won’t be graded).

- No make-up tests or quizzes will be given and ***no*** late homework (or other work) will be accepted.
- **Keys to success**: Do the homework and attend the class!
- Cheating will not be tolerated, but working together is encouraged.

Text, Slides & Notes

- No required text
  - but loads of recommended references
- Slides will be on the Moodle:
- New MOODLE for Computer Science Students

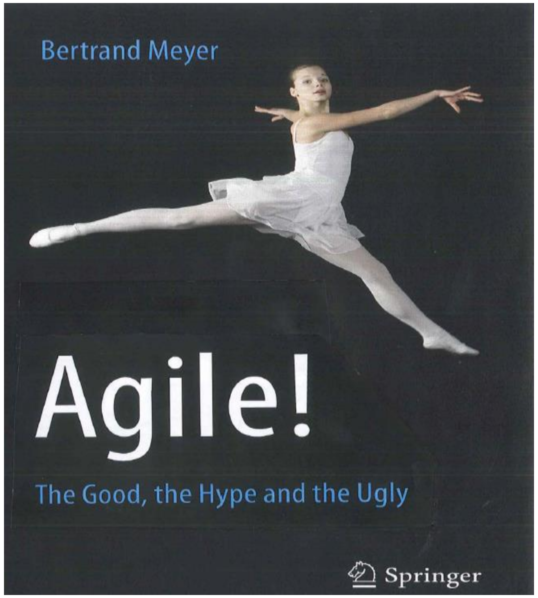
  https://csmoodle.ucd.ie/
- Logon using your UCD connect account
- Click on **COMP3005J**
- And enrolment key is **"2019COMP3005J"**

Reading

- **“Agile! The Good, the Hype and the Ugly”,** Bertrand Meyer, (Springer International Publishing, 2014, eBook ISBN: 978-3-319-05155-0, Softcover ISBN: 978-3-319-05154-3).

- **Bibliography** (Additional Material indicated in this course)

- Other useful resources:
  - “Agile Project Management for Dummies”, Mark C. Layton, MBA, CST, PMP, SCPM (Wiley, April 2012, ISBN: 978-1-118-23585-0)



Bibliography (1)

- [Anand site] Bachan Anand: Conscires site at agile.conscires.com.
- [Beck 05] Kent Beck, with Cynthia Andres: Extreme Programming explained — Embrace Change, Addison-Wesley, 2005 (Second Edition).
- [Boehm 81] Barry W. Boehm: Software Engineering Economics, Prentice Hall, 1981.
- [Chromatic 2003] Chromatic: Extreme Programming Pocket Guide, O'Reilly, 2003.
- [Cockburn 2001] Alistair Cockburn: Agile Software Development, Addison-Wesley, 2001.
- [Cockburn 2005] Alistair Cockburn: Crystal Clear — A Human-Powered Methodology for Small Teams, Addison Wesley, 2005.
- [Cunningham 2004] Ward Cunningham (interviewed by Bill Venners): The Simplest Thing That Could Possibly Work, at www.artima.com/intv/simplest.html.
- [Cohn 2006] Mike Cohn: Agile Estimating and Planning, Addison-Wesley, 2006.
- [Cohn 2010] Mike Cohn: Succeeding with Agile, Addison-Wesley, 2010.
- [Cohn Site] Mike Cohn: Succeeding with Agile site, www.mountaingoatsoftware.com.
- [DeMarco 1999] Tom DeMarco and Tim Lister: Peopleware: Productive Projects and Teams (Second Edition), Dorset House, 2001.
- [Fowler 1999] Martin Fowler: Refactoring: Improving the design of existing code, Addison-Wesley, 1999 .
- [Jeffries 2001] Ron Jeffries, Ann Anderson and Chet Hendrickson: Extreme Programming Installed, Addison-Wesley, 2001.
- [Jeffries site] Ron Jeffries: Xprogramming site at xprogramming.com.
- [Jobs 1998] Steve Jobs on Apple's resurgence (Andy Reinhardt), at http://allaboutstevejobs.com/sayings/stevejobsinterviews/bw98.php

Bibliography(2)

- [Poppendieck 2003] Mary and Tom Poppendieck: Lean Software Development — An Agile Toolkit, Addison-Wesley, 2003.
- [Poppendieck 2010] Mary and Tom Poppendieck: Leading Lean Software Development, Addison Wesley, 2010.
- [Royce 1970] Winston D. Royce: Managing the Development of Large Software Systems, in Proc. IEEE WESCON, 1970, pages 1-9, at www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf
- [Saunders] Gareth Saunders blog at blog.garethjmsaunders.co.uk
- [Schwaber 2004] Ken Schwaber: Agile Project Management with Scrum, Microsoft Press, 2004.
- [Shore 2008] James Shore and Shane Warden: The Art of Agile Development, O'Reilly, 2008.
- [Shore 2008a] James Shore: The Decline and Fall of Agile, blog article, 14 November 2008, at www.jamesshore.com/Blog/The-Decline-and-Fall-of-Agile.html.
- [Shore site] Web site for [Shore 2008] at jamesshore.com/Agile-Book.
- [Sutherland 2009] Jeff Sutherland: Self-Organization: The Secret Sauce for Improving your Scrum Team, video at www.youtube.com/watch?v=M1q6b9JI2Wc.
- [Sutherland 2012] Jeff Sutherland, Rini van Solingen and Telco Rustenberg: The power of Scrum, CreateSpace, 2012.
- [Sutherland 2013] Jeff Sutherland: Scrum: The Art of Doing Twice the Work in Half the Time, tutorial notes, 2013, available at jeffsutherland.com/CSMjsv16.pdf
- [Tomayko 03] James Tomayko James Herbsleb: How Useful Is the Metaphor Component of Agile Methods? A Preliminary Study, Report CMU-CS-03-152, School of Computer Science, Carnegie-Mellon University, June 2003.

Bibliography (3)

- [Wake site] Bill Wake: Exploring Extreme Programming site at xp123.com.
- [Wallace 2002] Doug Wallace, Isabel Raggett & Joel Aufgang: Extreme Programming for Web Projects, Addison-Wesley, 2002.
- [Waters site] Kelly Waters: All About Agile site, www.allaboutagile.com.
- [Wells site] Don Wells: Extreme Programming site, www.extremeprogramming.org.
- [Yourdon 2003] Ed Yourdon: Death March, 2nd edition. Prentice Hall, 2003.
- [Zave 1997] Pamela Zave and Michael Jackson: Four dark corners of requirements engineering, in ACM TOSEM (Transactions on Software Engineering and Methodology), vol. 6, no. 1, January 1997, pages 1-30.

Plagiarism

- When submitting homework quizzes and assignments, you should be aware that plagiarism is very serious.

- Any work you submit should be your own work.
  - Copying directly from the internet is not acceptable.
  - Copying your work from a classmate is not acceptable.

- I want to see what *you* know and how *you* would describe or understand the material.