

Latest update: January 31, 2005

## **PROJECT TOPICS FOR CIS6370 (COP6855)**

### **General remarks.**

Projects can be a research paper, based on analyzing two or three papers on some subject or an implementation or extension of some approach.

Projects can be individual or in a group of up to 4 people. Obviously I expect more from larger groups. Projects must be related to the course topics.

You can select a topic from my list or propose your own. In all cases please send me a one-page proposal describing your topic. Proposals must be approved by me.

Note that this is a 6000-level graduate course and I expect your reports to reflect that level. This is not an introductory, purely descriptive course, so your report conclusions should be justified precisely by argument, modeling or quoting previous work.

All of this means that your topic should be rather narrow. You cannot be precise or say anything new if you have a broad topic.

A possible organization for your report is :

1. **Introduction.** Brief description of objectives. Motivation. Previous related work. Approach (what will you find and how). Overview of the rest of the paper.

2. **Background.** What the reader needs to know to understand your paper. Assume the reader knows OO, UML, patterns, basic security concepts. Give references to where one can find more details.

3. **Results.** This is the most important part of your report. Use tables, OO models, patterns, architectural diagrams to develop your ideas and to prove your points.

4. **Conclusions.** What have we learned? How would you continue if you had more time?

5. **References.** Papers or books quoted in your report. No need for bibliography (list of related references). Use complete references: author, journal or conference, date, page numbers). My class notes are a publication and should be given as a reference whenever relevant.

**Appendix:** Copies of relevant papers (if hard to find or important to understand your report).

Look at *IEEE Computer* papers for examples of good writing for a report. Use clear, concise English, few adjectives. Relate your development and conclusions to the concepts discussed in class. A good report can become a conference or journal paper (it has happened many times in the past). It can also be the basis for a thesis or dissertation.

When reading vendors' material be wary of their claims, their products are always the best, the most secure, the fastest. Don't repeat this unless you have an independent proof.

Papers should be typewritten in double space. Include your e-mail and telephone number in the report in case I need to contact you because something is missing or unclear. Please

submit hard copies of your report; in some cases I may ask also for electronic copies. Remote FEEDS students may submit electronic versions.

Grading is based on originality, completeness, correctness, and relation to class subjects.

Projects related to my current research (indicated as \*) are specially encouraged. Look in <http://polaris.cse.fau.edu/~ed/pubs.html> to get a better idea of those topics.

### Topics for projects

--Study how to map security mechanisms across architectural levels. This is of high practical value because there are many uncoordinated products. A good try is the use of security patterns at different levels.

See <http://www.cse.fau.edu/~ed/secActivities.html> (DEXA'99 paper)

--Study the following papers:

M.S. Olivier, Towards a configurable security architecture, *Data & Knowledge Engineering*, Vol. 38, No. 2 (2001) , pages 121-145.

Lang, T., Fernandez, E.B., and Summers R., "A system architecture for compile-time actions in databases," *Proceedings of the ACM 1977 Annual Conference*, pp. 453-462, October 1977

Both use separation of security from data processing aspects. Develop a security pattern to include these two (and maybe other) cases.

--Survey the security of current hardware processor architectures (Pentium, Power PC, ...).

--Compare methods for decentralized security administration.

--\*Survey the status of security models for the business levels of web services (ebXML, BPEL, UDDI, WSDL,...). Or better, develop an approach to secure BPEL workflows.

--\*Patterns for security in VOIP (Voice over IP) networks. Consider wireless aspects.

--Study security for distributed systems such as Jini.

--Survey security aspects of genomic systems

--\*Security models for HIPAA or Sarbanes/Oxley regulations.

--Security in grid computing. Grid computing is done using web services.

--Secure electronic voting, including voting through the Internet

--\*Consider how to add security in distributed architectural patterns such as the Broker, Lookup, resource allocation patterns. I have a semi-finished paper on secure brokers.

- A paper proposed protecting relational databases by dispersing parts of the relations in different distributed nodes. Extend this approach to XML documents.
- Study security aspects of P2P systems.
- \*Survey models of trust, in particular to support wireless networks.
- \*Survey security using Aspect-Oriented design. Compare to my approach that uses patterns.
- Survey security in ad-hoc or sensor networks
- Hardware-based security (Palladium).
- Redo some of the cryptography patterns that have appeared and develop new ones.
- Security in the semantic web. Consider also semantic web services.
- Study spy malware. How do they operate? How to stop or destroy them?
- \*Survey location-aware security. Consider possible situations and define what data should be controlled.
- Common Criteria and web services.
- \*Security in wireless web services. Matching of wireless and web services standards for security.
- \*Access control of wireless devices. Assume that there is no centralized set of rules to control access to resources in the net.
- \*Wireless privacy—Effect of location. Relationship to P3P and EPL.
- \*Develop an access control model for controlling access to physical locations, e.g. rooms, shelves.
- \*Compare my security requirements approach based on use cases to other approaches using misuse cases or protecting assets.