Computer Systems Security – Survey/Experimental Projects

Please read the proposed projects and make your choice (survey or experimental project and topic) **before 15 November**. Starting Monday, 17 November please select your project by a) sending an email to the teacher, and b) visiting AISE Lab by Friday, 21 November, to discuss details and plan a roadmap.

1. Setup and Schedule

The goal of the projects is to deepen your knowledge in a particular domain that you select which is related to computer systems security. The project (see Section 2) you may work on may be:

- a **review paper** (where the deliverable is a 30-min presentation enhanced with examples), or
- an **experimental project** (where the deliverable is a 20-min slide presentation, and a 10-min demo).

The maximum grade is usually **8.0/10.0 for a review** and **10.0/10.0 for an experimental project,** with exceptions related to the complexity of the task.

You can work on your project survey during the next 7 weeks. For all projects, all related materials (slide presentation, any well-documented code, makefiles, testbench data and a brief README ascii file), should be submitted in a zip before the exam date.

All projects will be presented in AISE Lab on Jan 9, 16:00 – 22:00 (presentations & demonstrations)

For experimental projects, lab work concentrates in the first 5 weeks (before the Xmas holidays) and presentation during the remaining 3 weeks before the exam (after the Xmas holidays). Most experimental projects require lab presence (AISE Lab).

Note: For experimental projects, it is sometimes possible to form a group of up to 2 students and work on a project that naturally splits into two related subprojects. In this case, both students must work together on integration and present a common slide presentation and demo.

2. Project Types & Topics

There are two project types:

Review Paper [Max Grade 8.0/10.0, i.e., max 40 points contribution to final grade]

Your project surveys classic research related to. You need to review classic papers related to systems security protocols authentication, authorization, vulnerabilities, security and privacy, cryptography, or specific use cases or methodologies, and prepare a 30-minute presentation with 20-to-30 slides. The presentation should explain the problem and motivation, classify the approaches or techniques, and discuss results or possible future research directions. Paper reviews do not require programming work, but you must prepare a presentation that covers the subject extensively. To improve understanding of protocols, it is better to work on your own examples.

Suggested topics for Review Paper (Gray ones may ideally extend past student work):

1. Extended authorization protocols - Examples
2. Hardware CWEs & Vulnerabilities – Examples, e.g., embedded security
3. Different Cryptographic Protocols & Games
4. Hash Chains in security protocols. Example: Lamport's remote authentication, Guy Hawkes protocol
5. Hash Trees: protocols and their use. Example: Merkle trees
6. Hash-based Signatures: Example: Lamport's original scheme and extensions

7. Esoteric Languages
8. Blockchain with examples, e.g., cryptocurrencies
9. Secure Electronic Transactions
10. Security and safety-related standards in transportation, especially automotive. Examples: ISO 26262, ISA/IEC 62443, ISO/SAE 21434
11. Anomaly detection in automotive networks: specification-based, frequency-based (statistics), and machine-learning techniques
12. Social Engineering techniques
13. Security & Public Policy – examples, e.g., secure electronic voting protocols
14. Malware analysis & visualization
15. Quantum Computing and Post-Quantum Cryptography

Experimental Project [Max Grade 10.0/10.0, i.e., max 50 points contribution]

Projects focus on research/development related to information systems security and embedded security (hardware & software). Projects consist of implementing an idea, concept, design, framework, or tool related to systems security. The student must develop the system prototype, experiment on the platform, and prepare a 20-minute presentation (15-to-20 slides) that identifies the security problem, presents the attacks/solutions, compares with related work, and examines future work.

Suggested topics (Gray ones may ideally extend past student work):

1) Examine hardware fingerprints (PUFs) and the use of RNGs in security protocols (**in-lab-work**)

- Generate TRNG & PUFs, e.g., from uninitialized SRAMs
- Experiment with TRUE RNG hardware (TRNG chip, INFNOISE)
- Develop a fingerprint-related protocol and prepare a demo
- Examine their use in authentication protocols, e.g., generating session keys

2) RFID-related Protocols (**in-lab-work**)

- Learn about Secure Electronic Transactions (SET)
- Experiment with SET protocols using TLS & possibly Zymbit zymkey crypto IC

3) RFID smart cards: technology & attacks (**in-lab-work**)

- Examine tag architecture, security, and previous attacks on smart cards
- Discover innovative security protocols that relate to cheap RFID cards, such as Mifare Classic
- Build custom security protocols, use of multiple tags
- Integrate RFID tech to perform security challenges, enable vehicle diagnostics (OBD), or patch running code on embedded systems
- Use Proxmark3 to reverse engineer & attack, e.g., sniff, replay or clone different smartcards

4) CAN Bus: Security Protocols and Attacks - Real Time Issues (**in-lab-work**)

- Examine Security & Safety Standards in Automotive Systems
- Design attacks and examine/develop detection/protection (IDPS) mechanisms over CAN bus
- Develop a secure travel datalogger from malicious third-party exploits
- Experiment on embedded platform that connects ECUs to ECUSIM2000 (engine simulator)

5) Hardware Security Modules: Zymbit zymkey and ATECCx08 crypto devices (**in-lab-work**)

- Integrate a crypto device (ATECC608a or Zymbit zymkey) computer networks (TLS)

- Integrate a crypto device (ATECC608a or Zymbit zymkey) with CAN bus requests/replies
- Explore the Microchip cryptoauth world: provisioning & protocols (experiment with ATECC508 on Arduino, RPI or Dragonboard 410c)

6) ARM Trustzone (**in-lab-work**)

- Study ARM Trustzone sw development (secure, rich world), Cheri/ARM Morello architecture
- Examine ARM Trustzone design technologies and abstractions (e.g., Trustonic)
- Experiment with Microchip ATSAML11 using Microchip Studio, MPLAB
- Examine ARM Trustzone implementation on USB Armory (secure boot, etc)

7) Use Hack RF for wireless hacking (**in-lab-work**)

- Use GNUradio Companion and other tools to capture wireless communication patterns and perform attacks
- Consider Bluetooth (using BT devices), IR (using IR Anavi), TV signals etc

8) Add data privacy & security to an E-health application (**in-lab-work**)

- Use open source heartypatch device (a cardiac sensor)
- Develop secure biosensor driver/application that runs on 2-core ESP32
- Develop authentication, encryption, or anonymity protocols within an app that runs in FreeRTOS

9) Secure Probing via I2C, SPI and UART (**in-lab-work**)

- Examine how to design secure sensors (over I2C, SPI or UART)
- Examine the status of related patents etc
- Develop a prototype related to secure access to sensors
- Demonstrate probing using a prototype with probes (Aardvark, Beagle) or OBD dev kit

10) Secure network logs and actions (**in-lab-work**)

- Examine syslog and Secure Event Correlator for correlating network logs and honeypot technology
- Develop a framework with dynamic security rules (regular expressions) and actions on an embedded platform, e.g., Odroid XU4
- Examine real-time visualization, e.g., especially heatmaps

11) Network firewalls (**in-lab-work**)

- Examine network firewall and network packet capture technology, including kernel support
- Examine related bandwidth management techniques
- Examine methods, filters & languages for specifying rules, e.g., iptables, netfilter ufw, BPF, etc
- Examine distributed firewalls and related open issues

12) Design methodology for detection/protection of/from hardware trojans (**in-lab-work**)

- Examine methodology/tools for detecting sensitive data and locating possible hardware trojans
- Experiment with trojans on a network-on-chip firewall (prototype on FPGA)

13) Experiment with covert channels (**partly in-lab-work**)

- Discover interesting scenarios and implementations of covert channels
- Experiment with software channels, e.g., networking, audio
- Examine hardware channels, e.g., optical (optisense device and tomu/arduino led)

14) Examine and evaluate key loggers (**partly no in-lab-work**)

- Examine the design of different hacking mechanisms
- Experiment with user-level apps and Linux kernel rootkit modules
- Examine related hacking software (e.g., metasploit, openvas, etc) - experiment with VMs
- Use rubber ducky hardware dongle and other similar hacking devices from Hak5

15) Software vulnerabilities - stack & heap overflows (**in-lab-work**)

- Examine stack & heap overflow
- Study stack smashing attacks
- Study heap attacks on physical memory using pagemap analysis
- Develop a demo attack on a device driver that stores sensitive data sent from a biosensor in a statically or dynamically memory allocated array

16) Reverse Engineering (**no in-lab-work**)

- Describe security analysis methods and tools
- Example: Measure entropy for identifying keys, e.g., using binwalk, bindiff on images
- Experiment with Ghidra

17) Steganography (**no in-lab-work**)

- Steganography vs Cryptography
- Experiment with different steganography toolkits and libraries
- Design and implement related use cases

18) Secure Chat Protocols (**no in-lab-work**)

- Examine different secure chat & IRC protocols
- Compare the design of different secure chat protocols and evaluate software overheads

Lab presence & Periodic Reporting

You can work in the lab as often as necessary. Usually, 2 preferably consecutive days per week, 3-4 hours each time are fair enough for the projects. The available times for lab work, technical support, and progress report (before Xmas) are:

Monday, Tuesday, Wednesday, and Thursday: 14:00 to 18:00 at AISE Lab (lab work, technical support)

Friday: 16:00 to 20:00 at AISE Lab (lab work, technical support, and weekly progress report, if not communicated before)

3. Guidelines for Presentations (All projects)

Your presentation should be well structured and to the point. Clarity in the slides and graphics is very important. The font size should be 18 points or more for the class audience to be able to follow. You should speak clearly and avoid reciting. Time yourself in advance to avoid exceeding your 20 min slot.
The content of your slides must have the following structure:

- title page & student name(s),
- technical problem that your prototype aims to investigate or resolve,
- motivation by justifying why solving this problem is important,
- outline your experimental methodology **(for experimental projects only)**,
- report and explain the major results,

- survey existing solutions and compare results with previous related work,
- discuss future work, and
- provide a list of bibliography/references.

For the experimental methodology, you should:

- discuss the experimental prototype framework or design approach used for conducting experiments (e.g., implementation, instrumentation, measurement, simulation, or analytic modeling),
- clearly outline the set of experiments performed (threat model, assumptions, use case and scenarios),
- discuss your code and any improvements you made (with sanity/validation experiments or troubleshooting for evaluating correctness),
- analyze results and discuss conclusions of your experiments,
- explain how well the experiments helped resolve the technical problem by relating it to qualitative, and especially quantitative measurements (e.g., evaluate security overheads related to system or application efficiency metrics: latency, throughput, power consumption, enhance your presentation/demo with animations/real-time graphics), and
- identify additional development work (theoretical or experimental) for the future.

Bonus Points (for experimental work): compare your work with existing work, and/or conduct experiments that shed light on new innovative features.

4. Guidelines for Experimental Project Demonstrations

If you choose an experimental project, then immediately after your presentation, you must follow up with a demo of 10 minutes. Extra time, during or after the demo, will be available for questions from all.

Notice that for effective demonstration:

- Your demo should explain your experimental framework, including any input, your code or scripts, any compiling steps, and selective runs with results
- For enhancing the audience's experience:
    - Keep the demonstration simple
    - Practice showmanship and do troubleshooting. Everything should be ready to go
    - Relate the demo to your presentation
    - Get the audience involved, encourage questions
    - Enhance your demo with well-structured, real-time graphics
    - Present & demo in English!

5. Grading (Slides, Demo, Code) -

Title, Technical Problem & Motivation	10
Prototype Solution (Assumptions, Threat Model/Attack, Code or Scenario)	25
Results (Qualitative/Quantitative) & Comparison	25
Future Work & Bibliography	10
Structure, Clarity, Correctness, Timing	10
Demo Quality, Explanations (Scenario/Code), Knowhow, Effort/Collaboration, Innovation	20

(**Note: Submit (Slides, Demo, Code) before the deadline!)

Assessment will consider the following criteria:

- Problem Definition (15%): The clarity and significance of the research problems addressed in the thesis.
- Design Methodology (20%): The robustness and appropriateness of the methodology employed in conducting the research and achieving the results.
- Achievement and Significance (20%): The extent to which the thesis achieves its objectives and its significance to IoT.
- Originality and Innovation (20%): The level of originality and innovation demonstrated in the research ideas, approaches, and findings.
- Impact (15%): The potential impact of the thesis on advancing knowledge and practice in the IoT domain.
- Quality of Presentation (10%): The clarity, coherence, and effectiveness of the thesis presentation, including organization, writing style, and visual aids.