

Modulhandbuch Master of Science (M.Sc.)

IT-Sicherheit / Informationstechnik [PO22]

Stand: Sommersemester 2025

<https://informatik.rub.de/studium/studiengaenge/its/mits/>



Studienplan Master IT-Sicherheit/ Informationstechnik PO 22

| Nr | Modul | Umfang bzw. Mind. Umfang (CP) | Empfohlenes Semester | Bewertung |
|---------------------------|-----------------------------------|---|----------------------|-----------|
| Wahlpflichtbereich | | | | |
| 1 | Theorie der IT-Sicherheit ** | a * | 1-3 | benotet |
| 2 | Anwendungen der IT-Sicherheit *** | b * | 1-3 | benotet |
| 3 | Informatik **** | c * | 1-3 | benotet |
| 4 | Praktikum/ Projektarbeit ***** | 4 | 1-3 | unbenotet |
| 5 | Seminar ***** | 3 | 1-3 | benotet |
| Wahlbereich | | | | |
| 6 | Freie Wahlmodule ***** | ≥ 25 | 1-3 | unbenotet |
| Abschlussarbeit | | | | |
| 8 | Masterarbeit und Kolloquium | 27+3 | 4 | benotet |
| Summe: | | 120 | | |

#

* $a \geq 15, b \geq 15, c \geq 15, a+b+c \geq 58$

** Hier sind Module aus dem Wahlpflichtkatalog Theorie der IT-Sicherheit zu belegen. Die wählbaren Module sind im jeweils aktuellen Modulhandbuch aufgeführt.

*** Hier sind Module aus dem Wahlpflichtkatalog Anwendungen der IT-Sicherheit zu belegen. Die wählbaren Module sind im jeweils aktuellen Modulhandbuch aufgeführt.

**** Hier sind Module aus dem Wahlpflichtkatalog Informatik zu belegen. Die wählbaren Module sind im jeweils aktuellen Modulhandbuch aufgeführt.

***** Informationen zu den angebotenen Seminaren und Praktika finden Sie im Vorlesungsverzeichnis der RUB.

***** Hier können (nahezu) alle Veranstaltungen des Vorlesungsverzeichnisses der RUB, sowie Veranstaltungen im Rahmen der Universitätsallianz Ruhr gewählt werden. #

Angebotene Wahlpflichtmodule

| Lehrveranstaltung | Einheit | Umfang Modul (LP) | Semester | Bewertung |
|---|------------|-------------------|-------------------------------|-----------|
| Wahlpflichtmodule | | | | |
| Theorie der IT-Sicherheit | | | | |
| Proofs are programs | Informatik | 5 | WS (kein Angebot im WS 25/26) | benotet |
| Public Key Verschlüsselung | Informatik | 5 | WS | benotet |
| Quantum Information and Computation | Informatik | 5 | WS | benotet |
| Quantum Cryptography | Informatik | 5 | WS (kein Angebot im WS 25/26) | benotet |
| Symmetrische Kryptanalyse | Informatik | 5 | WS | benotet |
| Advanced Quantum Information and Computation | Informatik | 5 | SS | benotet |
| Authentische Schlüsselvereinbarung: Formale Modelle und Anwendungen | Informatik | 5 | SS | benotet |
| Foundations of Programming Languages, Verification, and Security | Informatik | 5 | SS | benotet |
| Fundamentals of Data Science | ETIT | 5 | SS | benotet |
| Kryptographische Protokolle | Informatik | 5 | SS | benotet |
| Provable Security - Promises and Misconceptions | Informatik | 5 | SS | benotet |
| Public Key Kryptanalyse 1 | Informatik | 5 | SS (nicht im SS 25) | benotet |
| Zero-Knowledge Proof Systems | Informatik | 5 | SS | benotet |
| Deep Learning | Informatik | 5 | Letztmalig WS 22/23 | benotet |
| Anwendungen der IT-Sicherheit | | | | |
| Aufbau eines Managementsystems für Informationssicherheit nach DIN ISO / 27001 | Informatik | 4 | WS/SS | benotet |
| Aktuelle Themen im Bereich der Internet-Sicherheit | Informatik | 5 | WS | benotet |
| Blockchain and Decentralized Security (ehemals Blockchain Security and Privacy) | Informatik | 5 | WS | benotet |
| Empirische IT-Sicherheitsforschung | Informatik | 5 | WS (kein Angebot im WS 25/26) | benotet |
| Human Aspects of Cryptography Adoption | Informatik | 5 | WS | benotet |
| Introduction to Cybercrime and Incident Response | Informatik | 4 | WS | benotet |
| Message Level Security | Informatik | 5 | WS | benotet |
| Microarchitectural Attacks and Defenses | Informatik | 5 | WS | benotet |
| Privacy Engineering, Data Governance and Usability | Informatik | 5 | WS | benotet |
| Software Security | Informatik | 9 | Letztmalig WS 23/24 | benotet |
| Software Security 1 | Informatik | 5 | WS | benotet |
| Advanced Automatic Testing | Informatik | 5 | SS | benotet |
| Developer Centered Security | Informatik | 5 | Letztmalig SS 24 | benotet |
| Digitale Souveränität | Informatik | 6 | SS | benotet |
| Menschliches Verhalten in der IT-Sicherheit | Informatik | 5 | SS | benotet |
| Mobile Network Security | Informatik | 5 | SS | benotet |
| Physical Attacks and Countermeasures | Informatik | 5 | SS | benotet |
| Processor Security | Informatik | 5 | SS | benotet |
| Programmanalyse | Informatik | 5 | SS | benotet |
| Software Protection | Informatik | 5 | SS | benotet |
| Software Security 2 | Informatik | 5 | SS | benotet |
| Software-Implementierung kryptographischer Verfahren | Informatik | 5 | SS | benotet |
| Informatik | | | | |
| Advanced Algorithms | Informatik | 9 | WS | benotet |
| Autonomous Vehicles and Artificial Intelligence Lab | Informatik | 5 | WS | benotet |
| Deep Learning | Informatik | 5 | WS | benotet |
| Deterministic Network Calculus | Informatik | 9 | WS | benotet |
| Energy-Aware Computing Systems | Informatik | 6 | WS | benotet |
| Formal Verification and Model Checking | Informatik | 5 | WS | benotet |
| Fundamentals of GPU Programming | ETIT | 5 | WS | benotet |

| | | | | |
|---|------------|---|---------------------|---------|
| Functional Programming | Informatik | 5 | WS | benotet |
| Komplexitätstheorie | Informatik | 9 | WS | benotet |
| Autonomous Vehicles and Artificial Intelligence | Informatik | 5 | SS | benotet |
| Effiziente Algorithmen | Mathematik | 9 | SS | benotet |
| Embedded Multimedia | ETIT | 6 | SS | benotet |
| Finite Fields: Theory and Algorithms | Informatik | 5 | SS | benotet |
| Highlights of Theoretical Computer Science | Informatik | 9 | SS | benotet |
| Machine Learning: Supervised Methods | Informatik | 6 | SS (nicht im SS 25) | benotet |
| Datenbanksysteme | Mathematik | 9 | Letztmalig WS 22/23 | benotet |
| Distributed System | Informatik | 5 | Letztmalig SS 23 | benotet |
| Knowledge Graphs | Informatik | 5 | Letztmalig SS 23 | benotet |
| Künstliche Neuronale Netze | Informatik | 6 | Letztmalig WS 22/23 | benotet |
| Nebenläufige Programmierung | Informatik | 5 | Letztmalig SS 23 | benotet |
| Web-Engineering | Bauing | 5 | Letztmalig SS 23 | benotet |
| Information Theory | Informatik | 5 | Letztmalig SS 23 | benotet |

Angebote Vertiefungsseminare

| Lehrveranstaltung | Einheit | Umfang Modul (LP) | Semester | Bewertung |
|---|------------|-------------------|------------------------|-----------|
| Vertiefungsseminare | | | | |
| Seminar Human Centered Security and Privacy | Informatik | 3 | WS/SS | benotet |
| Information Security Seminar | Informatik | 3 | WS/SS | benotet |
| Master-Seminar "Digitale Souveränität" | Informatik | 3 | WS/SS | benotet |
| Seminar Netz- und Datensicherheit | Informatik | 3 | WS/SS | benotet |
| Seminar Security Engineering | Informatik | 3 | WS/SS | benotet |
| Seminar Software and Internet Security | Informatik | 3 | WS/SS | benotet |
| Seminar Randomisierte Algorithmen | Informatik | 3 | WS/SS (nicht im SS 25) | benotet |
| Seminar zur symmetrischen Kryptographie | Informatik | 3 | WS/SS (nicht im SS 25) | benotet |
| Seminar on Security and Privacy of Ubiquitous Systems | Informatik | 3 | WS/SS | benotet |
| Seminar Automated Software Engineering | Informatik | 3 | WS/SS | benotet |
| Seminar zur Real World Cryptoanalysis | Informatik | 3 | WS | benotet |
| Seminar Mobile Network Security | Informatik | 3 | WS (nicht im WS 25/26) | benotet |
| Seminar in Advanced Automated Testing | Informatik | 3 | WS | benotet |
| Seminar Quantum Information and Computation (ehemals Quantum Cryptography) | Informatik | 3 | WS | benotet |
| Seminar Ressourceneffiziente Systemsoftwarekonzepte | Informatik | 3 | WS | benotet |
| Seminar on Applied Privacy and Anonymity | Informatik | 3 | SS | benotet |
| Seminar Quantum Algorithms | Informatik | 3 | SS (nicht im SS 25) | benotet |
| Seminar Perlen der Logik (ehemals Satisfiability) | Informatik | 3 | SS | benotet |
| Current topics in microarchitectural security | Informatik | 3 | SS | benotet |
| Seminar From Biological to Artificial Neural Networks | Informatik | 3 | SS | benotet |
| Seminar Mathematics and Computation | Informatik | 3 | SS | benotet |
| Seminar Implementation Security | Informatik | 3 | Letztmalig SS 23 | benotet |
| Perlen der theoretischen Informatik (ehemals Grenzen in der theoretischen Informatik) | Informatik | 3 | Letztmalig WS 23/24 | benotet |
| Master-Seminar Developer Centered Security | Informatik | 3 | Letztmalig SS 24 | benotet |
| Master Seminar Security and Privacy for Mobile Systems | Informatik | 3 | Letztmalig WS 23/24 | benotet |
| Seminar on Current Topics for Systems Security and Privacy | Informatik | 3 | Letztmalig SS 24 | benotet |

Angebote Praktika/Projekte

| Lehrveranstaltung | Einheit | Umfang Modul (LP) | Semester | Bewertung |
|--|------------|-------------------|----------------------------------|-----------|
| Projekt Netz- und Datensicherheit | Informatik | 4 | WS/SS | unbenotet |
| Forschungspraktikum Human-Centred Security | Informatik | 4 | WS/SS | unbenotet |
| Praktikum zur Hackertechnik (Hackerpraktikum) | Informatik | 4 | WS/SS | unbenotet |
| Master-Forschungspraktikum (Laborstudien) Human-Centred Security | Informatik | 4 | WS/SS | unbenotet |
| Master-Praktikum Wireless Physical Layer Security | ETIT | 4 | WS/SS | unbenotet |
| Research in Information Security (Master Project) | Informatik | 4 | WS/SS | unbenotet |
| Master-Praktikum Reverse-Engineering Security Features | Informatik | 4 | WS/SS (kein Angebot im WS 25/26) | unbenotet |
| Research in Internet Security | Informatik | 4 | WS/SS | unbenotet |
| Research in Software Security | Informatik | 4 | WS/SS | unbenotet |
| Projekt Eingebettete Sicherheit | Informatik | 4 | WS/SS (nicht im SS 25) | unbenotet |
| Research in Ubiquitous Systems | Informatik | 4 | WS/SS | unbenotet |
| Software Testing via Fuzzing | Informatik | 4 | WS/SS | unbenotet |
| Advanced Research in Microarchitectural Security | Informatik | 4 | WS (nicht im WS 25/26) | unbenotet |
| Lab Course: Challenging Problems in Reinforcement Learning | Informatik | 4 | WS | unbenotet |
| Practical Course on Machine learning Security | Informatik | 4 | WS | unbenotet |
| Praktikum TLS Implementierung | Informatik | 4 | WS | unbenotet |
| Praktikum ARM Processors for Embedded Cryptography | Informatik | 4 | WS | unbenotet |
| Praktikum Implementing Post-Quantum Standards and Challenges | Informatik | 4 | WS | unbenotet |
| Practical Course Traffic Analysis Attacks | Informatik | 4 | WS | unbenotet |
| Introductory project in microarchitectural security | Informatik | 4 | SS | unbenotet |
| Practical Course on Mobile Network Security | Informatik | 4 | SS | unbenotet |
| Practical Course on Blockchain Security | Informatik | 4 | SS | unbenotet |
| Practical IoT Hacking | Informatik | 4 | SS | unbenotet |
| Projekt Research in Security Engineering | Informatik | 4 | SS | unbenotet |
| Praktische Kryptanalyse von symmetrischen Chiffren | Informatik | 4 | SS (nicht im SS 25) | unbenotet |
| Embedded Firmware Fuzzing | Informatik | 4 | SS | unbenotet |
| Praktikum Seitenkanalangriffe | Informatik | 4 | Letztmalig WS 22/23 | unbenotet |
| Developer Centered Security (Projekt) | Informatik | 4 | Letztmalig SS 24 | unbenotet |
| Creating Mystery Twister Crypto Challenges | Informatik | 3 | SS | unbenotet |

Abkürzungen:

SS: Sommersemester

WS: Wintersemester

CP: Creditpoints

ETIT: Fakultät für Elektrotechnik und Informationstechnik

Baulng: Fakultät für Bau- und Umweltingenieurwissenschaften

MODULHANDBUCH

Übersicht der Module

IT-Sicherheit / Informationstechnik - Master (1-Fach, PO 2022)

Wahlpflichtbereich

Advanced Algorithms

Autonomous Vehicles and Artificial Intelligence

Autonomous Vehicles and Artificial Intelligence Lab

Deep Learning

Deterministic Network Calculus

Effiziente Algorithmen

Embedded Multimedia

Energy-Aware Computing Systems

Finite Fields: Theory and Algorithms

Formal Verification and Model Checking

Functional Programming

Fundamentals of GPU Programming

Highlights of Theoretical Computer Science [M.Sc]

Information Theory

Komplexitätstheorie

Machine Learning: Supervised Methods (kein Angebot im SS 25)

Advanced Automatic Testing

Aktuelle Themen im Bereich der Internet-Sicherheit

Aufbau eines Managementsystems für Informationssicherheit nach DIN ISO / 27001

Blockchain and Decentralized Security

Digitale Souveränität

Empirische IT-Sicherheitsforschung (kein Angebot im WS 25/26)

Human Aspects of Cryptography Adoption

Introduction to Cybercrime and Incident Response

Menschliches Verhalten in der IT-Sicherheit

Message Level Security

Microarchitectural Attacks and Defenses

Mobile Network Security

Physical Attacks and Countermeasures

Privacy Engineering, Data Governance and Usability

Processor Security

Programmanalyse [M.Sc.]

Software Protection
Software Security 1 [M.Sc.]
Software Security 2
Software-Implementierung kryptographischer Verfahren
Advanced Quantum Information and Computation
Authentische Schlüsselvereinbarung: Formale Modelle und Anwendungen
Foundations of Programming Languages, Verification, and Security
Fundamentals of Data Science
Kryptographische Protokolle
Proofs are programs [M.Sc.]
Provable Security - Promises and Misconceptions
Public Key Kryptanalyse 1 [M.Sc.] (nicht im SoSe 25)
Public Key Verschlüsselung
Quantum Cryptography (kein Angebot im WS 25/26)
Quantum Information and Computation [M.Sc.]
Symmetrische Kryptanalyse
Zero-Knowledge Proof Systems
Master Praktikum/Projektarbeit IT-Sicherheit
Vertiefungsseminar (M.Sc. IT-Sicherheit)

Wahlbereich

Freie Wahlmodule

Abschlussarbeit

Masterarbeit und Kolloquium (ITS)

| | | | | | |
|--|------------------------|--------------------------|---|---------------------------------|---------------------------------------|
| Titel des Moduls: Advanced Algorithms Advanced Algorithms | | | | | |
| Modul-Nr./Code | Credits 9 CP | Workload 270 h | Semester siehe Prüfungsordnung / see Examination regulations | Turnus Wintersemester | Dauer 1 Semester |
| Lehrveranstaltungen Advanced Algorithms (212029) | | | Kontaktzeit 6 SWS (90 h) | Selbststudium 180 h | Gruppengröße 40 Studierende |
| Unterrichtssprache Englisch | | | Teilnahmevoraussetzungen keine | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Maike Buchin Lehrende: Prof. Maike Buchin | | | | | |
| Verwendung des Moduls M.Sc. Computer Science M.Sc. Angewandte Informatik M.Sc. Mathematik M.Sc. IT-Sicherheit / Informationstechnik | | | | | |
| Vorkenntnisse Empfohlen: Erwartet werden grundlegende Kenntnisse zu Algorithmenentwurf und -analyse wie sie aus dem Bachelorstudium bekannt sind. | | | | | |
| Lernziele (learning outcomes) <ul style="list-style-type: none"> • Fortgeschrittene Entwurfsmethoden für Algorithmen • Fortgeschritten Analysemethoden für Algorithmen • Kenntnis weiterer Datenstrukturen und Methoden zum Entwurf von Datenstrukturen • Anwendung der gelernten Methoden auf neue Probleme | | | | | |
| Inhalt In der Vorlesung betrachten wir fortgeschrittene Themen der Algorithmik. Nach einer kurzen Wiederholung bekannter Inhalte betrachten wir vor allem Graphalgorithmen, Approximationsalgorithmen und FPT-Algorithmen sowie exakte Algorithmen für NP-schwere Probleme. Ebenfalls betrachten wir einige neue und bekannte Datenstrukturen und deren Analyse. Die betrachteten Probleme dabei sind sowohl kombinatorisch, graphentheoretisch also auch geometrisch. | | | | | |
| Lehrformen Vorlesung (als Folien- und Tafelvortrag) und Übungen, in denen die vorgestellten Inhalte vertieft werden | | | | | |
| Prüfungsformen Mündliche (20-30 Minuten) oder schriftliche Modulabschlussprüfung (120 Minuten) (wird zu Semesterbeginn bekannt gegeben) | | | | | |
| Voraussetzungen für die Vergabe von Credits Bestandene Modulabschlussprüfung und erfolgreiche Teilnahme an Übungen | | | | | |
| Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 9/97: M.Sc. Computer Science | | | | | |

9/105: M.Sc. Angewandte Informatik

9/91: M.Sc. IT-Sicherheit / Informationstechnik [PO 22]

9/84: M.Sc. IT-Sicherheit / Informationstechnik [PO 20]

Titel des Moduls: Advanced Automatic Testing
Advanced Automatic Testing

| | | | | | |
|---|------------------------|-------------------------|---|---------------------------------|---------------------------------------|
| Modul-Nr./Code | Credits 5 CP | Workload 150h | Semester siehe Prüfungsordnung / see examination regulations | Turnus Sommersemester | Dauer 1 Semester |
| Lehrveranstaltungen Advanced Automatic Testing (211067) | | | Kontaktzeit 60 h (4 SWS) | Selbststudium 90h | Gruppengröße 20 Studierende |
| Unterrichtssprache Englisch | | | Teilnahmevoraussetzungen Keine | | |

Modulbeauftragte/r und hauptamtlich Lehrende

Modulbeauftragte/r: Prof. Dr. Flavio Toffalini
 Prof. Dr. Yannic Noller
 Lehrende: Prof. Dr. Flavio Toffalini
 Prof. Dr. Yannic Noller

Verwendung des Moduls

Vorkenntnisse

Die Studierenden sollten über Vorkenntnisse im Bereich der System- und Softwaresicherheit verfügen. Frühere Kurse in Systemprogrammiersprachen (z. B. C und C++) und ein Mindestmaß an gegnerischer Sicherheit.

Lernziele (learning outcomes)

Die Studierenden erwerben Kenntnisse über automatische Testverfahren mit Schwerpunkt auf praktischen Aspekten bei Sicherheitsproblemen. Darüber hinaus wird der Kurs die neuesten Entwicklungen auf diesem Gebiet untersuchen.

Inhalt

Dieser Kurs befasst sich mit den Grundlagen des automatischen Testens (Fuzzing) in der Softwaresicherheit und bietet Masterstudenten die Möglichkeit, ihre Fachkenntnisse in diesem Bereich zu vertiefen.

Der Kurs bietet einen umfassenden Überblick über automatisches Testen und behandelt grundlegende Konzepte wie White-Box-, Grey-Box- und Black-Box-Tests, Standard- und moderne Code-Explorationstechniken sowie fortgeschrittene Fehlererkennung mit logikbasierten Orakeln. Jede Lektion konzentriert sich auf einen bestimmten Aspekt der Disziplin, wodurch schrittweise ein vollständiges Verständnis des Themas aufgebaut wird und die Studenten mit den Fähigkeiten ausgestattet werden, neue Konzepte in diesem Bereich unabhängig zu erforschen.

Der Kurs deckt die folgenden Themen ab, die je nach Bedarf an die Anforderungen der Klasse angepasst werden:

- Einführung in das Fuzzing-Paradigma: Black-box-, Grey-box- und White-box-Ansätze
- Arten des Abdeckungsfeedbacks
- Heuristiken für die Codeuntersuchung (z. B. Mutatoren, Metastrategien, Seed-Auswahl, Grammatiken)
- Grundlegende Konzepte der symbolischen Ausführung und des konkollischen Testens
- Fehlererkennung und Replikation
- Fehlerorakel (z. B. Sanitizer, differenzielle Tests)
- Verwaltung von Fuzzing-Kampagnen
- Domänenspezifisches Testen (z. B. Kernel, virtuelle Geräte, Bibliotheken, IoT-Geräte)

Übungen ergänzen die Vorlesungen, indem sie praktische Erfahrungen mit den unterrichteten Prinzipien bieten. Diese Übungen sind unerlässlich für die Entwicklung des praktischen Wissens und der Problemlösungsfähigkeiten, die für die Abschlussprüfung erforderlich sind, sowie für ein gründliches Verständnis

des im Kurs vorgestellten Materials.

Lehrformen

Der Kurs kombiniert wöchentliche Vorlesungen und Laborsitzungen. Während der Vorlesungen werden die Studierenden ermutigt, sich mit dem Dozenten auszutauschen und sich an der Lösung einfacher Aufgaben zu beteiligen, um ihre Lernerfahrung zu verbessern.

Prüfungsformen

Schriftliche Modulabschlussprüfung (120 Minuten). In den Übungen können Bonuspunkte erlangt werden.

Voraussetzungen für die Vergabe von Credits

Die Studierenden müssen die schriftliche Abschlussprüfung bestehen.
Sie können Bonuspunkte für das Lösen der in den Übungen präsentierten Aufgaben erhalten.

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/97: M.Sc. Computer Science

5/105: M.Sc. Angewandte Informatik

5/91: M.Sc. IT-Sicherheit / Informationstechnik

5/99: M.Sc. IT-Sicherheit/ Netze und Systeme

Titel des Moduls: Advanced Quantum Information and Computation
Advanced Quantum Information and Computation

| | | | | | |
|--|------------------------|--------------------------|---|---|---------------------------------------|
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung / see examination regulations | Turnus Tentatively every summer semester | Dauer 1 Semester |
| Lehrveranstaltungen Advanced Quantum Information and Computation (211003) | | | Kontaktzeit 4 SWS (60 h) | Selbststudium 90 h | Gruppengröße 30 Studierende |
| Unterrichtssprache Englisch | | | Teilnahmevoraussetzungen | | |

Modulbeauftragte/r und hauptamtlich Lehrende

Modulbeauftragte/r: [Prof. Dr. Michael Walter](#),
 Dr. Simon Schmidt
 Lehrende: Prof. Dr. Michael Walter

Verwendung des Moduls

M.Sc. Computer Science
 M.Sc. IT-Sicherheit / Informationstechnik
 M.Sc. IT-Sicherheit / Netze und Systeme
 M.Sc. Angewandte Informatik
 M.Sc. Mathematik
 M.Sc. Physik

Vorkenntnisse

Successful participation of Quantum Information and Computation (or an equivalent course). No background in physics is required.

Lernziele (learning outcomes)

You will learn fundamental concepts, algorithms, and results in quantum information and computation that go beyond a first course. You will be prepared for a research or thesis project in this area.

Inhalt

This topical course is meant as a follow-up to our introductory course Quantum Information and Computation and is aimed at students interested in deepening their knowledge in this area. We plan to cover selected topics in quantum information and computation, e.g. how to model quantum channels, analyze nonlocal games, design quantum algorithms and cryptographic protocols, and obtain insights into which problems are easy and which are likely hard even for quantum computers. Students interested in a Bachelor's or Master's project in quantum information, computing, cryptography, etc. are particularly encouraged to participate.

Lehrformen

Vorlesung mit Übung

Prüfungsformen

Final exam (the format will depend on the number of participants).

Voraussetzungen für die Vergabe von Credits

Passed final exam

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/97: M.Sc. Computer Science

5/91: M.Sc. IT-Sicherheit / Informationstechnik [PO 22]

5/84: M.Sc. IT-Sicherheit / Informationstechnik [PO 20]

5/99: M.Sc. IT-Sicherheit / Netze und Systeme [PO 22]

5/96: M.Sc. IT-Sicherheit / Netze und Systeme [PO 20]

5/105: M.Sc. Angewandte Informatik

| | | | | | |
|--|------------------------|-----------------|---------------------------------|---------------------------------|------------------------------------|
| Titel des Moduls: Aktuelle Themen im Bereich der Internet-Sicherheit | | | | | |
| Modul-Nr./Code | Credits 5 CP | Workload | Semester | Turnus Wintersemester | Dauer 1 Semester |
| Lehrveranstaltungen 211099 - Aktuelle Themen im Bereich der Internet-Sicherheit | | | Kontaktzeit | Selbststudium | Gruppengröße Studierende |
| Unterrichtssprache Deutsch | | | Teilnahmevoraussetzungen | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Jörg Schwenk Lehrende: Prof. Jörg Schwenk | | | | | |
| Verwendung des Moduls | | | | | |
| Vorkenntnisse Keine | | | | | |
| Lernziele (learning outcomes) In der Vorlesung werden ausgewählte Themen der IT-Sicherheit behandelt, die vom Lehrstuhl für Netz- und Datensicherheit in den letzten Jahren publiziert wurden. Es werden unter anderem folgende Themen behandelt: <ul style="list-style-type: none"> • Portable Document Flaws • Overview over Cryptographic Modelling with the Example of Messaging • 0-RTT and Tor • Padding Oracles • Racocon • Breaking Microsoft RMS 2020 • IPsec-Bleichenbacher • DEMONS: DNS-Poisoning by Exhaustive Misappropriation of Network Sockets • DOM • XS Leaks • UI Redressing <p>Neben den Systemen selbst werden dabei auch publizierte Angriffe auf diese Systeme besprochen; die Studierenden stellen selbst wissenschaftliche Überlegungen zur Verbesserung der Sicherheit an.</p> | | | | | |
| Inhalt Studierende verfügen nach erfolgreichem Abschluss des Moduls über ein umfassendes Verständnis der aktuellen Forschungsthemen im Bereich der Internet-Sicherheit. Sie haben die neuesten Angriffe und Sicherheitsmechanismen kennengelernt. Zusätzlich wissen Sie, wie man mit Sicherheitsschwachstellen korrekt umgeht und wie man diese an den Hersteller meldet. Durch die wissenschaftsnahen Themen haben die Studierenden Einblicke in die Forschung im Bereich der Internetsicherheit gekriegt, wodurch sie sich auch auf ihre potentielle Forschungsrolle vorbereitet haben. | | | | | |
| Lehrformen Vorlesung | | | | | |
| Prüfungsformen Schriftliche Klausur (120 Minuten) | | | | | |
| Voraussetzungen für die Vergabe von Credits Bestandene Modulabschlussprüfung | | | | | |

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

Titel des Moduls: Aufbau eines Managementsystems für Informationssicherheit nach DIN ISO / 27001

| | | | | | |
|---|------------------------|--------------------------|---|---------------------------------|------------------------------------|
| Modul-Nr./Code | Credits 4 CP | Workload 120 h | Semester siehe Prüfungsordnung | Turnus Sommersemester | Dauer 1 Semester |
| Lehrveranstaltungen Aufbau eines Managementsystems für Informationssicherheit nach DIN ISO / IEC 27001 (211021) | | | Kontaktzeit 45 h | Selbststudium 75 h | Gruppengröße Studierende |
| Unterrichtssprache Deutsch | | | Teilnahmevoraussetzungen Keine | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Professur für Systemsicherheit Lehrende: Dr.-Ing. Sebastian Uellenbeck | | | | | |
| Verwendung des Moduls M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Netze und Systeme | | | | | |
| Vorkenntnisse Vorkenntnisse über Systemsicherheit und Netzsicherheit z. B. aus den Vorlesungen Systemsicherheit 1 und Netzsicherheit 1 | | | | | |
| Lernziele (learning outcomes) Die Studierenden haben ein fundiertes Verständnis über den Aufbau eines ISMS nach ISO 27001 und kennen die notwendigen Schritte, um ein Unternehmen zur Zertifizierungsreife zu begleiten. Studierende können eigenständig neue Probleme analysieren und neue Lösungsmöglichkeiten entwickeln. Sie können mit diesem Verständnis mit ihren Kollegen über ISO/IEC 27001 diskutieren und auftretende Probleme im Gespräch korrekt klassifizieren. | | | | | |
| Inhalt Die Lehrveranstaltung vermittelt fokussiert Inhalte aus der ISO/IEC 27001 Auditorensicht. Dazu ist folgende Gliederung geplant: <ul style="list-style-type: none"> • Zielsetzung • Prinzipien und Terminologien • Auditprinzipien gemäß ISO 19011:2011 Richtlinien • ISO 19011 • ISO 27001:2013 Dokumentation • Auditvorbereitung: Pre-Audit Meeting und Auditpläne • Vorbereitung von Checklisten • Audittechniken • Auditorenpräsentationen • Auditergebnisse und Abschlusstreffen • Abweichungen, Bericht der Beobachtungen und Folgemaßnahmen • Folgemaßnahmen <p>Weitergehend werden technische Lösungsmittel besprochen, die auf dem Weg zur ISO 27001 Zertifizierung hilfreich sein können. Hierzu zählen unter anderem Security Information and Event Management Systeme (SIEM) und Identity Management Systeme (IdM).</p> | | | | | |
| Lehrformen Vorlesung mit Übung (Blockveranstaltung in den Semesterferien Anmeldung über sysec@rub.de) | | | | | |

Prüfungsformen

schriftliche Modulabschlussprüfung (90 Minuten)

Voraussetzungen für die Vergabe von Credits

bestandene schriftliche Modulabschlussprüfung

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

4/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

4/99: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 22]

| | | | | | |
|--|------------------------|--------------------------|--|---------------------------------|------------------------------------|
| Titel des Moduls: Authentische Schlüsselvereinbarung: Formale Modelle und Anwendungen | | | | | |
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung | Turnus Sommersemester | Dauer 1 Semester |
| Lehrveranstaltungen Au-then-ti-sche Schlüs-sel-ver-ein-ba-rung: For-ma-le Mo-del-le und An-wen-dun-gen (211038) | | | Kontaktzeit 60 h | Selbststudium 90 h | Gruppengröße Studierende |
| Unterrichtssprache Deutsch | | | Teilnahmevoraussetzungen Keine | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Jörg Schwenk Lehrende: Prof. Dr. Jörg Schwenk | | | | | |
| Verwendung des Moduls M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Netze und Systeme | | | | | |
| Vorkenntnisse Grundkenntnisse Kryptographie Empfehlung: Durcharbeiten der ersten 40 Folien vom Skript Kryptographie I von Prof. Alexander May | | | | | |
| Lernziele (learning outcomes) Die Studierenden verstehen die Besonderheit kryptographischer Protokolle, bei denen nicht mehr ein Algorithmus im Vordergrund steht, sondern die Interaktion verschiedener Einheiten. Sie kennen die wichtigsten Konzepte bzgl. der beweisbaren Sicherheit von Protokollen. Die wichtigsten Bausteine kryptographischer Protokolle werden behandelt, so dass die Studierenden in der Lage sind, direkt in die wissenschaftliche Literatur zu diesem Thema einzusteigen. | | | | | |
| Inhalt Das Modul bietet eine Einführung in das Gebiet der kryptographischen Protokolle, die den Einsatz bekannter und neuer Verfahren der Kryptographie in der Kommunikation zwischen mehreren Instanzen beschreibt. Hierbei wird sowohl Wert auf die Beschreibungen als auch auf die Sicherheit gelegt. Die Vorlesung umfasst folgende Themen: Kryptographische Grundlagen (Kurze Wiederholung der Wahrscheinlichkeitstheorie, Informationstheorie, etc.) Beweisbare Sicherheit Analyse von Schlüsselaustauschprotokollen, mit besonderem Fokus auf praktische Beispielprotokolle (wie TLS oder SSH) Die Zusammenstellung ist nicht fest und kann nach Absprache mit den Hörern auch geändert werden. | | | | | |
| Lehrformen Vorlesung mit Übung | | | | | |
| Prüfungsformen schriftlich, 120 Minuten | | | | | |
| Voraussetzungen für die Vergabe von Credits Bestandene Modulabschlussprüfung | | | | | |
| Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) | | | | | |

5/91: M.Sc. IT-Sicherheit/ Informationstechnik

5/99: M.Sc. IT-Sicherheit/ Netze und Systeme

Titel des Moduls: Autonomous Vehicles and Artificial Intelligence
Autonomous Vehicles and Artificial Intelligence

| | | | | | |
|--|------------------------|--------------------------|---|---------------------------------|---------------------------------------|
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung / see examination regulations | Turnus Sommersemester | Dauer 1 Semester |
| Lehrveranstaltungen Autonomous Vehicles and Artificial Intelligence (211044) | | | Kontaktzeit 4 SWS (60 h) | Selbststudium 90 h | Gruppengröße 25 Studierende |
| Unterrichtssprache Englisch | | | Teilnahmevoraussetzungen keine | | |

Modulbeauftragte/r und hauptamtlich Lehrende

Modulbeauftragte/r: Prof. Thorsten Berger
 Lehrende: Prof. Dr. Thorsten Berger, Dr. Sven Peldszus

Verwendung des Moduls

B.Sc. Informatik [bis SS 23]
 B.Sc. IT-Sicherheit [bis SS 23]
 B.Sc. Angewandte Informatik [bis SS 23]
 M.Sc. Computer Science
 M.Sc. Angewandte Informatik
 M.Sc. IT-Sicherheit/Informationstechnik
 M.Sc. IT-Sicherheit/Netze und Systeme [bis SS 23]

Vorkenntnisse

Die Vorlesung Software Engineering oder eine vergleichbare Veranstaltung, Programmiererfahrungen z.B. im Rahmen anderer Lehrveranstaltungen.

Lernziele (learning outcomes)

- Verständnis der Anforderungen an autonome Fahrzeuge
- Verständnis der Architektur von autonomen Fahrzeugen
- Fähigkeit, ein selbstfahrendes Auto mit ROS2 zu bauen
- Verstehen und Anwenden der Qualitätssicherung für autonome Fahrzeuge

Inhalt

Autonomes Fahren ist die Zukunft der individuellen Mobilität und alle großen Hersteller arbeiten an vollautonomen Fahrzeugen. Während es für die einzelnen Probleme des autonomen Fahrens robuste und gute Lösungen gibt, liegt die größte Herausforderung in deren Integration. Insgesamt stellt die Software eines autonomen Fahrzeugs das größte Problem dar. Daher liegt der Schlüssel für selbstfahrende Fahrzeuge darin, die Software richtig zu machen. In diesem Kurs werden wir die verschiedenen Aspekte von selbstfahrenden Fahrzeugen sowie die Bedeutung und Anwendung von künstlicher Intelligenz in diesem Bereich untersuchen. Der Kurs wird sich hauptsächlich auf die folgenden Themen konzentrieren:

- Anforderungen an autonome Fahrzeuge

- Architektur von autonomen Fahrzeugen
- Betriebssysteme und Frameworks für Robotersysteme
- Spezifikation und Implementierung von autonomen Fahrzeugen auf Basis von ROS2
- Künstliche Intelligenz für autonome Fahrzeuge
- Simulation von autonomen Fahrzeugen
- Lokalisierung und Wahrnehmung
- Missionsplanung
- Qualitätssicherung für autonome Fahrzeuge In der Vorlesung werden die notwendigen theoretischen Grundlagen vermittelt und die Inhalte in Übungen durch den Bau eines selbstfahrenden Roboters praktisch angewendet.

Lehrformen

Vorlesung mit Übungen

Prüfungsformen

Mündliche Modulabschlussprüfung (30 Minuten)

Voraussetzungen für die Vergabe von Credits

Bestandene Modulabschlussprüfung und erfolgreiche Teilnahme an den Übungen

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/97: M.Sc. Computer Science

5/105: M.Sc. Angewandte Informatik

5/91: M.Sc. IT-Sicherheit/Informationstechnik [PO 22]

5/84: M.Sc. IT-Sicherheit/Informationstechnik [PO 20]

Titel des Moduls: Autonomous Vehicles and Artificial Intelligence Lab
Autonomous Vehicles and Artificial Intelligence Lab

| | | | | | |
|--|------------------------|-------------------------|---|---------------------------------|------------------------------------|
| Modul-Nr./Code | Credits 5 CP | Workload 150h | Semester siehe Prüfungsordnung / see examination regulations | Turnus Wintersemester | Dauer 1 Semester |
| Lehrveranstaltungen Autonomous Vehicles and Artificial Intelligence Lab (212035) | | | Kontaktzeit 4 SWS (60h) | Selbststudium 90h | Gruppengröße Studierende |
| Unterrichtssprache Englisch | | | Teilnahmevoraussetzungen - | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr Thorsten Berger Lehrende: Prof. Dr. Thorsten Berger Dr. Sven Peldszus | | | | | |
| Verwendung des Moduls M.Sc. Computer Science M.Sc. Angewandte Informatik M.Sc. IT-Sicherheit/Informationstechnik | | | | | |
| Vorkenntnisse Empfohlen Teilnahme an der Vorlesung Autonomous Vehicles and Artificial Intelligence (211044) Programmiererfahrung in C++ oder Python (z.B. als Teil von anderen Kursen) Teilnahme an der Vorlesung Software Engineering (212000) oder einer vergleichbaren Lehrveranstaltung | | | | | |
| Lernziele (learning outcomes) Wissen - relevante theoretische Kenntnisse über künstliche Intelligenz und autonome Fahrzeuge erläutern können Fertigkeiten und Fähigkeiten - Anforderungen an autonome Fahrzeuge definieren und validieren - eine Architektur für autonome Fahrzeuge erstellen - ein selbstfahrendes Auto mit ROS2 bauen - Management und Integration von künstlicher Intelligenz in komplexe, softwareintensive Systeme - Organisation eines Teams und seines Entwicklungsprozesses für ein komplexes, softwareintensives System - Qualitätssicherung für autonome Fahrzeuge durchführen - Erstellen der Dokumentation des Entwicklungsprozesses und der Artefakte, die für eine Zertifizierung nach den ISO-Normen für Straßenfahrzeuge benötigt werden - professionell mit Gruppenmitgliedern und Stakeholdern kommunizieren (in Wort und Schrift) | | | | | |

Inhalt

Autonomes Fahren ist die Zukunft der individuellen Mobilität, und alle großen Hersteller arbeiten an vollständig autonomen Fahrzeugen. Während es für die einzelnen Probleme des autonomen Fahrens robuste und gut erforschte Lösungen gibt, liegt die größte Herausforderung in deren Integration. Insgesamt stellt die Software eines autonomen Fahrzeugs das größte Problem dar. Daher liegt der Schlüssel für selbstfahrende Fahrzeuge darin, die Software richtig zu gestalten.

In diesem Kurs werden wir die verschiedenen Aspekte von selbstfahrenden Fahrzeugen sowie die Bedeutung und Anwendung von künstlicher Intelligenz in diesem Bereich anhand der Entwicklung eines selbstfahrenden Rennwagens praktisch studieren. Zu diesem Zweck werden die Teilnehmer mit ROS2-basierten Modellautos arbeiten. Ziel ist es, den Studierenden praktische Erfahrungen bei der Entwicklung eines autonomen Rennwagens und der Organisation des Entwicklungsprozesses zu vermitteln.

Lehrformen

Die wichtigste Lernsequenz des Kurses ist ein großes Praxisprojekt. Das Projekt wird in Gruppen durchgeführt, die iterativ einen autonomen Rennwagen entwickeln und dabei theoretisches Wissen über autonomes Fahren und Softwareentwicklung anwenden und festigen. Um das Lernen zu unterstützen, basiert das Autonomous Vehicles and Artificial Intelligence Lab auf seminarähnlichen Vorlesungen, die eine Plattform für Feedback und weitere Informationen bieten. Auf der Grundlage der gesammelten Informationen aktualisieren und verfeinern die Studierenden ihre Lösungen für ein autonomes Rennauto. Die kontinuierliche Reflexion über Praxis und Theorie wird durch die laufende Erstellung eines abschließenden Projektberichts parallel zur Entwicklung des Rennwagens unterstützt, in dem die Studierenden über ihr eigenes Lernen im Kurs, die Art und Weise, wie sie und ihr Team ihren Entwicklungsprozess angehen, und ihre technischen Lösungen reflektieren. Die Studenten erhalten regelmäßiges Feedback und Anleitung, um ihr Lernen zu unterstützen.

Prüfungsformen

Die Endnote wird auf der Grundlage der Teilnahme an der Entwicklung des selbstfahrenden Rennwagens, schriftlicher Projektberichte, des entwickelten autonomen Rennwagens und einer mündlichen Präsentation der Gruppenergebnisse ermittelt. Die Einzelnoten werden aus der Bewertung der individuellen und gruppenbezogenen Ergebnisse gebildet.

Voraussetzungen für die Vergabe von Credits

Eine aktive Beteiligung an der Entwicklung eines autonomen Rennwagens, regelmäßiger Besuch der seminarähnlichen Vorlesungen und erfolgreiche Erbringung aller Leistungen.

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/97: M.Sc. Computer Science

5/105: M.Sc. Angewandte Informatik

5/91: M.Sc. IT-Sicherheit/Informationstechnik [PO 22]

5/84: M.Sc. IT-Sicherheit/Informationstechnik [PO 20]

Titel des Moduls: Blockchain and Decentralized Security
Blockchain and Decentralized Security

| | | | | | |
|---|------------------------|--------------------------|---|---------------------------------|---------------------------------------|
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung / see examination regulations | Turnus Wintersemester | Dauer 1 Semester |
| Lehrveranstaltungen Blockchain and Decentralized Security (212007) | | | Kontaktzeit 4 SWS (60 h) | Selbststudium 90 h | Gruppengröße 30 Studierende |
| Unterrichtssprache Englisch | | | Teilnahmevoraussetzungen Pre-requisites: Intro to Crypto 1 and 2, System Security, Network Security 1. | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Ghassan Karame Lehrende: Prof. Dr. Ghassan Karame | | | | | |
| Verwendung des Moduls M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Netze und Systeme M.Sc. Computer Science | | | | | |
| Vorkenntnisse Background in system security, network security, cryptographic primitives (encryption methods, signatures, MACs, hash functions), and principles of communication networks is required. | | | | | |
| Lernziele (learning outcomes) Upon completion of this course, students are expected to be able to: <ol style="list-style-type: none"> 1. Reason about the security and privacy definitions of decentralized systems. 2. Explain the security of blockchains in light of the state of the art reported attacks. 3. Reason about possible network security and cryptographic countermeasures to deter attacks on decentralized platforms (blockchains and distributed platforms). 4. Explain best security/privacy practices to strengthen the security of existing blockchains and existing distributed learning platforms, and extract relevant lessons for the design of next-generation decentralized technologies. | | | | | |
| Inhalt The main objective of the course is to provide a comprehensive overview of the security and privacy of decentralized technologies. Course participants will be also introduced to the basic security and privacy provisions of existing popular blockchains, and will be exposed to the state-of-the-art attacks and threats reported against existing systems/deployments. The participants will also reason on the effectiveness of combining network-level security primitives, with novel cryptographic primitives to deter attacks on payment systems and on the security of federated learning and distributed learning. | | | | | |
| Lehrformen Übung mit Vorlesung | | | | | |
| Prüfungsformen Schriftliche Modulabschlussprüfung (120 Minuten) | | | | | |
| Voraussetzungen für die Vergabe von Credits Bestandene schriftliche Modulabschlussprüfung. | | | | | |

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/91: Master IT-Sicherheit | Informationstechnik [PO 22]

5/84: Master IT-Sicherheit | Informationstechnik [PO 20]

5/99: Master IT-Sicherheit | Netze und Systeme [PO 22]

5/96: Master IT-Sicherheit | Netze und Systeme [PO 20]

5/97: Master Computer Science

| | | | | | |
|---|------------------------|--------------------------|---|---------------------------------|---------------------------------------|
| Titel des Moduls: Deep Learning Deep Learning | | | | | |
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung / see examination regulations | Turnus Wintersemester | Dauer 1 Semester |
| Lehrveranstaltungen Deep Learning (212018) | | | Kontaktzeit 4 SWS (60 h) | Selbststudium 90 h | Gruppengröße 50 Studierende |
| Unterrichtssprache Englisch | | | Teilnahmevoraussetzungen | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Asja Fischer Lehrende: Prof. Dr. Asja Fischer | | | | | |
| Verwendung des Moduls M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Netze und Systeme [Bis WS 22/23] M.Sc. Angewandte Informatik M.Sc. Computer Science | | | | | |
| Vorkenntnisse Grundkenntnisse der Linearen Algebra und Wahrscheinlichkeitstheorie sind von Vorteil. | | | | | |
| Lernziele (learning outcomes) Die Vorlesung hat das Ziel, einen Einblick in dieses Gebiet zu vermitteln. Zu Beginn werden die grundlegenden Begriffe und Konzepte des maschinellen Lernens eingeführt. Im weiteren Verlauf wird auf verschiedene neuronale Netze, Gradienten-basierte Optimierungsverfahren und generative Modelle eingegangen. | | | | | |
| Inhalt Deep Learning ist ein Untergebiet des maschinellen Lernens, welches in den letzten Jahren zu Durchbrüchen in zahlreichen Anwendungsgebieten (wie z.B. in der Objekt- und Spracherkennung und der maschinellen Übersetzung) geführt hat. Deep Learning Methoden finden unter anderem Anwendung im Bereich IT Security. | | | | | |
| Lehrformen Vorlesung und Übung | | | | | |
| Prüfungsformen Schriftliche Modulabschlussprüfung (120 Minuten) | | | | | |
| Voraussetzungen für die Vergabe von Credits Bestandene schriftliche Modulabschlussprüfung. | | | | | |
| Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 5/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22] 5/84: M.Sc. IT-Sicherheit/ Informationstechnik [PO 20] 5/105: M.Sc. Angewandte Informatik 5/ 97: M.Sc. Computer Science | | | | | |

Titel des Moduls: Deterministic Network Calculus**Deterministic Network Calculus**

| | | | | | |
|--|------------------------|--------------------------|---|---------------------------------|------------------------------------|
| Modul-Nr./Code | Credits 9 CP | Workload 270 h | Semester siehe Prüfungsordnung / see examination regulations | Turnus Wintersemester | Dauer 1 Semester |
| Lehrveranstaltungen Deterministic Network Calculus (211054) | | | Kontaktzeit 6 SWS (90 h) | Selbststudium 180 h | Gruppengröße Studierende |
| Unterrichtssprache Englisch | | | Teilnahmevoraussetzungen | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Steffen Bondorf Lehrende: Prof. Dr. Steffen Bondorf | | | | | |
| Verwendung des Moduls M.Sc. Computer Science M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. Angewandte Informatik | | | | | |
| Vorkenntnisse Mathematik (Funktionsanalyse), Computernetze / Verteilte Systeme | | | | | |
| Lernziele (learning outcomes) Nach erfolgreichem Abschluss des Moduls werden die Studierenden in der Lage sein, <ul style="list-style-type: none">• komplexe, vernetzte Systeme als deterministische Warteschlangensysteme zu modellieren,• worst-case Leistungsanalysen von bestehenden Systemen bzw. Modellen durchzuführen,• die Herausforderungen bei der Leistungsdimensionierung von geplanten Systemen zu verstehen, dabei die Wirkungsweise zentraler Mechanismen in Computernetzen anhand des Network Calculus zu erklären,• die vorgestellten Verfahren gegeneinander abzugrenzen und auf wissenschaftliche Fragestellungen anzuwenden. | | | | | |
| Inhalt Verteilte Systeme sind heutzutage allgegenwärtig, und ihre Vernetzung ist von grundlegender Bedeutung für die kontinuierliche Verbreitung und damit Verfügbarkeit von Daten. Die Bereitstellung von Daten in Echtzeit ist einer der wichtigsten nichtfunktionalen Aspekte, den sicherheitskritische Netze gewährleisten müssen. Die formale Verifizierung der Datenkommunikation im Hinblick auf die worst-case Deadlines ist grundlegend für die Zertifizierung von neu entwickelten x-by-Wire-Systemen. Diese Verifizierung erlaubt den Start von Flugzeugen, das Lenken von Autos ohne mechanische Verbindung und den Betrieb sicherheitskritischer Industrieanlagen. Daher wurden verschiedene Methoden für die worst-case Modellierung und Analyse von Echtzeitsystemen entwickelt. Eine davon ist der Deterministische Network Calculus (DNC), eine vielseitige Technik, die in verschiedenen Bereichen wie Paketvermittlung, Task Scheduling, System on Chip, softwaredefinierte Netzwerke, Netzwerke in Rechenzentren und Netzwerkvirtualisierung eingesetzt werden kann. DNC ist eine Methode zur Ableitung deterministischer Schranken für zwei der vorrangigsten Leistungsmetriken in Kommunikationssystemen: <ul style="list-style-type: none">• die Ende-zu-Ende-Verzögerung von Datenflüssen und• der Speicherplatz, den ein Server benötigt, um alle eingehenden Daten in einer Warteschlange zu puffern. | | | | | |
| Lehrformen Vorlesung mit Übung | | | | | |

Prüfungsformen

Mündliche Modulabschlussprüfung

Voraussetzungen für die Vergabe von Credits

Bestandene mündliche Modulabschlussprüfung.

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/97: M.Sc. Computer Science

5/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

5/84: M.Sc. IT-Sicherheit/ Informationstechnik [PO 20]

5/105: M.Sc. Angewandte Informatik

| | | | | | |
|---|------------------------|--------------------------|---|---------------------------------|---------------------------------------|
| Titel des Moduls: Digitale Souveränität Digital Sovereignty | | | | | |
| Modul-Nr./Code | Credits 6 CP | Workload 180 h | Semester siehe Prüfungsordnung / see examination regulations | Turnus Sommersemester | Dauer 1 Semester |
| Lehrveranstaltungen Digitale Souveränität (211059) | | | Kontaktzeit 4 SWS (60 h) | Selbststudium 120 h | Gruppengröße 25 Studierende |
| Unterrichtssprache Deutsch oder Englisch | | | Teilnahmevoraussetzungen keine | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Karola Marky Lehrende: Prof. Dr. Karola Marky | | | | | |
| Verwendung des Moduls M.Sc. IT-Sicherheit / Informationstechnik M.Sc. IT-Sicherheit / Netze und Systeme M.Sc. Computer Science | | | | | |
| Vorkenntnisse empfohlen: Einführung in die Usable Security and Privacy | | | | | |
| Lernziele (learning outcomes) Studierende kennen verschiedene Definitionen, Kontexte und Use Cases für Digitale Souveränität sowie Beeinflussungsmechanismen moderner Digitalprodukte. Die Studierenden können selbstständig neue Use Cases analysieren und bewerten. | | | | | |
| Inhalt In dieser Vorlesung erlangen die Studierenden ein Verständnis von Digitaler Souveränität im heutigen Zeitalter. Dabei werden verschiedene Themenblöcke bearbeitet. Zunächst gibt die Vorlesung einen Grundüberblick über die Bandbreite Digitaler Souveränität und Wechselwirkungen innerhalb der Gesellschaft. Anschließend werden Designprinzipien und Use Cases im Kontext der breiten Bevölkerung, Organisationen und Staaten erläutert, darunter das Teilen von Daten im digitalen Raum, IT-Sicherheit in Organisationen, und E-Democracy. Vorlesungsbegleitend findet ein Projekt statt, welches die Studierenden in Gruppen bearbeiten und so durch „Hands-On“ lernen, bestimmte Szenarien selbstständig zu bewerten. | | | | | |
| Lehrformen Vorlesung mit Übung (Projekt) | | | | | |
| Prüfungsformen Mündliche Abschlussprüfung | | | | | |
| Voraussetzungen für die Vergabe von Credits Bestehen der mündlichen Abschlussprüfung; für die Qualität der Durchführung des zu bearbeitenden Projekts werden Bonuspunkte vergeben | | | | | |
| Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 6/91: M.Sc. IT-Sicherheit/Informationstechnik [PO 22] | | | | | |

6/84: M.Sc. IT-Sicherheit/Informationstechnik [PO 20]

6/99: M.Sc. IT-Sicherheit/Netze und Systeme [PO 22]

6/96: M.Sc. IT-Sicherheit/Netze und Systeme [PO 20]

6/97: M.Sc. Computer Science

| | | | | | |
|--|------------------------|--------------------------|---|---------------------------------|------------------------------------|
| Titel des Moduls: Effiziente Algorithmen | | | | | |
| Modul-Nr./Code | Credits 9 CP | Workload 270 h | Semester siehe Prüfungsordnung | Turnus Sommersemester | Dauer 1 Semester |
| Lehrveranstaltungen Effiziente Algorithmen (150320 + 150321) | | | Kontaktzeit 90 h | Selbststudium 180 h | Gruppengröße Studierende |
| Unterrichtssprache Deutsch | | | Teilnahmevoraussetzungen | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: PD Dr. Daniela Kacso Lehrende: PD Dr. Daniela Kacso | | | | | |
| Verwendung des Moduls M.Sc. Angewandte Informatik M.Sc. IT-Sicherheit/ Informationstechnik | | | | | |
| Vorkenntnisse Die Inhalte der Veranstaltung "Datenstrukturen" bzw. "Informatik 2". | | | | | |
| Lernziele (learning outcomes) Nach dem erfolgreichen Abschluss des Moduls: Die Studierenden <ul style="list-style-type: none"> • kennen, wählen aus und nutzen grundlegende Datenstrukturen und Graphenalgorithmen • sind in der Lage Analysetechniken (Korrektheitsbeweise und Laufzeitanalyse) zu erläutern und zu beurteilen • können auch bei praktischen Problemen entscheiden, welche der vermittelten Methoden/Algorithmen/Datenstrukturen anwendbar sind und diese nach Effizienz (insb. Laufzeit der Algorithmen) bewerten • können konkrete Anwendungsprobleme modellieren und bei Bedarf diese Algorithmen weiter entwickeln | | | | | |
| Inhalt Die Lehrveranstaltung kann sowohl in das Gebiet der praktischen als auch in das Gebiet der theoretischen Informatik eingeordnet werden. Die zentralen Themen sind die Folgenden: <ul style="list-style-type: none"> • Berechnung kürzester Pfade in Digraphen • Berechnung eines maximalen Flusses in einem Transportnetzwerk • Berechnung einer optimalen Lösung bei einem Zuordnungsproblem (auch Matching-Problem genannt) Darüber hinaus beschäftigen wir uns mit Anwendungen dieser grundlegenden Probleme. | | | | | |
| Lehrformen Vortrag der Lehrenden in der Vorlesung, Gruppenarbeit in den Übungen | | | | | |
| Prüfungsformen Schriftliche Modulabschlussprüfung (120 Minuten) | | | | | |
| Voraussetzungen für die Vergabe von Credits Bestandene schriftliche Modulabschlussprüfung | | | | | |

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

9/105: M.Sc. Angewandte Informatik

9/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

9/84: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

| | | | | | |
|---|------------------------|--------------------------|--|---------------------------------|------------------------------------|
| Titel des Moduls: Embedded Multimedia | | | | | |
| Embedded Multimedia | | | | | |
| Modul-Nr./Code | Credits 6 CP | Workload 180 h | Semester | Turnus Sommersemester | Dauer 1 Semester |
| Lehrveranstaltungen Embedded Multimedia | | | Kontaktzeit 60 h | Selbststudium 120 h | Gruppengröße Studierende |
| Unterrichtssprache Deutsch | | | Teilnahmevoraussetzungen Keine | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr.-Ing. Rainer Martin Lehrende: Dr. Wolfgang Theimer | | | | | |
| Verwendung des Moduls Master IT-Sicherheit/ Informationstechnik | | | | | |
| Vorkenntnisse Kenntnis der Programmiersprache C/C++ Objektorientierte Programmierung Grundlagen der Signalverarbeitung | | | | | |
| Lernziele (learning outcomes) Die Studierenden erwerben grundlegende Fertigkeiten für das Systemdesign, die Implementierung, sowie die Integrations- und Testphase von Multimedialösungen im Bereich Embedded Systems. Sie sind befähigt, Hardware- und Softwarearchitekturen von eingebetteten Multimediasystemen zu bewerten. Sie sammeln anhand einer Linux-basierten Plattform Programmiererfahrungen und lösen in einem Projektteam eine Aufgabe aus dem Bereich der Multimediakommunikation. | | | | | |
| Inhalt Die Lehrveranstaltung vermittelt die Grundlagen zur Durchführung von Entwicklungsarbeiten im Bereich der eingebetteten Systeme, und hat den Fokus Multimediatechnologien. Zu Beginn der Vorlesung wird eine kurze Einführung in die Entwicklungsprozesse wie System-Engineering, Softwareentwicklung und Testvorgehen gegeben, um die Projektteams methodisch vorzubereiten. Anschließend werden grundlegende Hardware- und Softwarearchitekturen von Embedded Systems präsentiert, um sie zu befähigen, Lösungskonzepte einordnen zu können. Der Fokus der Lehrveranstaltung liegt danach in der detaillierten Analyse einer eingebetteten Plattform am Beispiel des Raspberry Pi. Die Nutzung der Prozessorplattform und der Peripheriekomponenten wird anhand der plattformübergreifenden Entwicklungsumgebung Qt Creator unter C/C++ vertieft. Im Rahmen der praktischen Umsetzung in einem Projektteam erwerben die Studierenden die Fähigkeiten, gemeinsam ein Entwicklungsproblem zu strukturieren, ein Lösungskonzept zu entwickeln, und unter Zuhilfenahme von existierenden Softwaremodulen zu einer Gesamtlösung zu integrieren. Die Herangehensweise an die Problemstellung und die Lösung sind vom Projektteam zu dokumentieren und abschließend allen Teilnehmern zu präsentieren. | | | | | |
| Lehrformen Vorlesung mit integrierten Übungen | | | | | |
| Prüfungsformen schriftlich, 120 Minuten | | | | | |
| Voraussetzungen für die Vergabe von Credits Praxisprojekt - Mündliche Prüfung | | | | | |
| Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 6/91: M.Sc. IT-Sicherheit/ Informationstechnik | | | | | |

Titel des Moduls: Empirische IT-Sicherheitsforschung (kein Angebot im WS 25/26)
Empirical Security Research (no offer in WS 25/26)

| | | | | | |
|---|------------------------|--------------------------|---|---------------------------------|------------------------------------|
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung | Turnus Wintersemester | Dauer 1 Semester |
| Lehrveranstaltungen Empirische IT-Sicherheitsforschung (212036) | | | Kontaktzeit 60 h | Selbststudium 90 h | Gruppengröße Studierende |
| Unterrichtssprache Englisch | | | Teilnahmevoraussetzungen | | |

Modulbeauftragte/r und hauptamtlich Lehrende
 Modulbeauftragte/r: Prof. Dr. M. Angela Sasse
 Lehrende: Prof. Dr. M. Angela Sasse, Annalina Buckmann, M.A.

Verwendung des Moduls
 M.Sc. IT-Sicherheit/Informationstechnik
 M.Sc. IT-Sicherheit/Netze und Systeme

Vorkenntnisse
 Grundkenntnisse der IT-Sicherheit, Grundkenntnisse der Human-Centred Security

Lernziele (learning outcomes)
 Students will learn fundamentals of IT Security Research and research planning: general research ethics considerations, and security-specific considerations (Menlo Report), and how to address them in study designs. Framing of study questions, selection of valid methods and metrics (qualitative and quantitative). Selection of data analysis methods and supporting tools. Communication limitations and recommendations. Documenting and applying lessons learnt.

Inhalt
 IT security researchers have traditionally focused on identifying vulnerabilities in IT systems and infrastructure, and develop solutions for the ones they find. In practice, their effectiveness is usually determined by compliance with standards or guidelines, or audits. But what is a valid scientific approach to determine how vulnerable a system is? How can we measure whether a solution has improved security? The course will introduce foundations and methods for conducting empirical security research, covering both technology-based research (e.g. vulnerability scans, penetration testing, reverse engineering) and human-based research (laboratory and online experiments, survey-based studies, interview-based studies, field studies, ethnography, participatory action research, inclusive security engagements).

Lehrformen
 - Lecture
 - The practical exercises will include teaching forms such as group and project work.

Prüfungsformen
 Oral Exam

Voraussetzungen für die Vergabe von Credits
 Passed Oral Exam

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)
 5/91: M.Sc. IT-Sicherheit/Informationstechnik [PO 22]
 5/84: M.Sc. IT-Sicherheit/Informationstechnik [PO 20]
 5/99: M.Sc. IT-Sicherheit/Netze und Systeme [PO 22]

Titel des Moduls: Energy-Aware Computing Systems
Energy-Aware Computing Systems

| | | | | | |
|---|------------------------|--------------------------|---|---------------------------------|---------------------------------------|
| Modul-Nr./Code | Credits 6 CP | Workload 180 h | Semester siehe Prüfungsordnung / see examination regulations | Turnus Wintersemester | Dauer 1 Semester |
| Lehrveranstaltungen Energy-Aware Computing Systems (212030) | | | Kontaktzeit 4 SWS (60 h) | Selbststudium 120 h | Gruppengröße 20 Studierende |
| Unterrichtssprache Englisch | | | Teilnahmevoraussetzungen keine | | |

Modulbeauftragte/r und hauptamtlich Lehrende

Modulbeauftragte/r: Prof. Dr.-Ing. Timo Hönig
 Lehrende: Prof. Dr.-Ing. Timo Hönig

Verwendung des Moduls

M.Sc. Computer Science
 M.Sc. IT-Sicherheit/ Informationstechnik
 M.Sc. Angewandte Informatik

Vorkenntnisse

Lernziele (learning outcomes)

Studierende, die die Vorlesung und die Übungen erfolgreich besucht haben, haben die Lernziele verfolgt und die unten aufgeführten Kompetenzen erworben. Die Studierenden können

- die Bedeutung von elektrischer Energie als Betriebsmittel für Rechensysteme verstehen
- Trade-off-Entscheidungen im Hinblick auf ein effizientes Systemdesign (d.h. Energiebedarf vs. Leistung), insbesondere von Betriebssystemen, treffen
- modellieren den Energiebedarf für einzelne synchrone und asynchrone Operationen
- Strategien zur Reduzierung des Energiebedarfs für Software-Aktivitäten auf der Grundlage spezifischer Hardware-Eigenschaften (z. B. Ruhezustände) anwenden
- Software auf kritische Abschnitte, die einen hohen Energiebedarf verursachen, analysieren.

Inhalt

Elektrische Energie ist die wichtigste Betriebsressource für Computersysteme. Obwohl der Energiebedarf von Computern an sich eine unsichtbare Systemeigenschaft ist, sind die Auswirkungen des Energiebedarfs allgegenwärtig und in verschiedenen Erscheinungsformen offensichtlich. Als praktische Beispiele dienen plötzliche Systemausfälle (d.h. Systemzusammenbrüche) und wiederkehrende Standard-Systemoperationen (d.h. Energiemanagement). Die Vorlesung befasst sich mit dem Entwurf energiebewusster Computersysteme und konzentriert sich auf die folgenden Themen:

- Leistungs- und Energiemanagement
- Energiebuchhaltung
- Analyse des Energiebedarfs
- energiebewusste Betriebssystem-Architektur
- Hardware-Energiemanagement (z.B. DVFS, Drosselung, Ruhezustände)
- Wärmemanagement
- Speicher- und Dateisysteme
- Speicherverwaltung
- Netzwerk, drahtlose Kommunikation und Protokolle
- Energiebewusste Server/Cluster

- Compiler-Optimierungen und Code-Umwandlung
- Anzeigetechnik
- Stromnetz

Die Vorlesung ist mit den Übungen durch Forschungsarbeiten verbunden. Die Studierenden lesen die Papiere zur Vorbereitung auf die Vorlesung. Von dort aus bilden die Forschungspapiere die Grundlage für die Diskussion und den Ausgangspunkt für die Aufgabenstellungen der Übungen. Im Rahmen der Übungen wenden die Studierenden Konzepte und Strategien aus den Forschungsarbeiten auf Systeme an und bewerten die Auswirkungen auf die Energieeffizienz des Systems.

Lehrformen

Die Vorlesung wird in Form eines Seminars abgehalten. Forschungsarbeiten zum energiebewussten Rechnen und Systemdesign werden von den Studierenden vorbereitet und in den Sitzungen diskutiert und analysiert. Zusätzlich vermittelt die Vorlesung theoretisches Wissen über grundlegende Konzepte zu den einzelnen Themen.

Im Rahmen der Übungen wenden die Studierenden ihr erworbenes Wissen an, indem sie Systemsoftware und Systemkonfigurationen zur Verbesserung der Energieeffizienz anpassen. Die Ergebnisse analysieren sie durch Leistungs- und Energiebedarfsauswertungen.

Prüfungsformen

Mündliche Modulabschlussprüfung (30 Minuten)

Voraussetzungen für die Vergabe von Credits

Bestandene Modulabschlussprüfung und erfolgreiche Teilnahme an den Übungen.

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

6/97: M.Sc. Informatik

6/105: M.Sc. Angewandte Informatik

6/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

6/84: M.Sc. IT-Sicherheit/ Informationstechnik [PO 20]

| | | | | | |
|--|------------------------|--------------------------|---|---------------------------------|------------------------------------|
| Titel des Moduls: Finite Fields: Theory and Algorithms Finite Fields: Theory and Algorithms | | | | | |
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung / see examination regulations | Turnus Sommersemester | Dauer 1 Semester |
| Lehrveranstaltungen Finite Fields: Theory and Algorithms (211058) | | | Kontaktzeit 4SWS (60h) | Selbststudium 90 h | Gruppengröße Studierende |
| Unterrichtssprache Englisch | | | Teilnahmevoraussetzungen | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Dr. Christof Beierle Lehrende: Dr. Christof Beierle | | | | | |
| Verwendung des Moduls M.Sc. Computer Science M.Sc. IT-Sicherheit / Informationstechnik | | | | | |
| Vorkenntnisse Lineare Algebra (z.B. aus "Mathematik 1" aus dem Bachelor-Studium), Grundkenntnisse in der Programmierung mit Python oder Sage, Interesse an Mathematik | | | | | |
| Lernziele (learning outcomes) Die Studierenden erhalten ein grundlegendes Verständnis der mathematischen Grundlagen endlicher Körper, ihrer Konstruktionen und ihrer algorithmischen Aspekte und sind in der Lage, mit endlichen Körpern unter Verwendung eines Computeralgebrasystems wie Sage zu arbeiten. | | | | | |
| Inhalt Endliche Körper haben zahlreiche Anwendungen in der Informatik, insbesondere in der Kryptographie und Codierungstheorie. Dieser Kurs ist eine Einführung in die Theorie der endlichen Körper und ihre algorithmischen Aspekte, beginnend mit den algebraischen Grundlagen. Themen: Algebraische Grundlagen, Polynomringe und Ideale, Euklidischer Algorithmus für Polynome, irreduzible Polynome, Erweiterungskörper, Minimalpolynome, Zerfällungskörper, Existenz und Eindeutigkeit endlicher Körper, Finden von primitiven Elementen der multiplikativen Gruppe, Frobenius-Automorphismus, Spur und Norm, quadratische Gleichungen über endlichen Körpern, Berlekamps Algorithmus zur Faktorisierung von Polynomen über kleinen endlichen Körpern, Permutationspolynome. | | | | | |
| Lehrformen Vorlesung mit Übung | | | | | |
| Prüfungsformen Schriftliche Modulabschlussprüfung (120 Minuten) | | | | | |
| Voraussetzungen für die Vergabe von Credits Bestandene schriftliche Modulabschlussprüfung | | | | | |
| Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 5/97: M.Sc. Computer Science | | | | | |

5/91: M.Sc. IT-Sicherheit / Informationstechnik [PO 22]

5/84: M.Sc. IT-Sicherheit / Informationstechnik [PO 20]

Titel des Moduls: Formal Verification and Model Checking**Formal Verification and Model Checking**

| | | | | | |
|---|------------------------|-------------------------|--|---------------------------------|---------------------------------------|
| Modul-Nr./Code | Credits 5 CP | Workload 150h | Semester siehe Prüfungsordnung / see Examination Regulations | Turnus Wintersemester | Dauer 1 Semester |
| Lehrveranstaltungen Formal Verification and Model Checking (212041) | | | Kontaktzeit 60h (4 SWS) | Selbststudium 90 | Gruppengröße 40 Studierende |
| Unterrichtssprache Englisch | | | Teilnahmevoraussetzungen Wenn bereits das frühere Modul "Model Checking" in einem vorherigen Semester absolviert wurde, ist ein Ablegen der Prüfung zum Modul "Formal Verification and Model Checking" ausgeschlossen. | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Nils Jansen Lehrende: Prof. Dr. Nils Jansen | | | | | |
| Verwendung des Moduls M.Sc. Computer Science M.Sc. Angewandte Informatik M.Sc. IT-Sicherheit/Informationstechnik | | | | | |
| Vorkenntnisse Grundkenntnisse der Automatentheorie. | | | | | |
| Lernziele (learning outcomes) Nach Abschluss dieses Kurses werden die Teilnehmer in der Lage sein <ul style="list-style-type: none">• Praktische Verifikationsprobleme mit modernen Werkzeugen wie SAT/SMT Solvern und Model Checkern wie PRISM oder Storm zu lösen.• Algorithmen zu verstehen, die modernen SAT- und SMT-Solvern wie Resolution, CDCL und CDCL(T) zugrunde liegen.• formale Logiken wie LTL oder CTL und die algorithmischen Implikationen der Modellprüfung anhand solcher Eigenschaften zu verstehen.• SAT- und SMT-Methoden mit Problemen aus der klassischen Planung, der Programmverifikation, der Modellprüfung und der probabilistischen Inferenz zu verbinden.• Situationen zu erkennen, in denen die Anwendung von Model Checking und Verifikationstechniken für Spezifikation und Analyse nützlich sein kann. Arbeiten Sie an innovativer Forschung, die maschinelles Lernen und künstliche Intelligenz mit den strengen Techniken der formalen Verifikation kombiniert. | | | | | |
| Inhalt Komplexe digitale Systeme sind in unserem Leben immer präsenter und wirken sich immer stärker aus, so dass die formale Überprüfung der Korrektheit dieser Systeme von entscheidender Bedeutung ist. So darf beispielsweise das Programm eines Raumschiffs nicht abstürzen, und ein Netzwerksystem sollte auch dann noch funktionieren, wenn ein Server ausfällt. Um dies zu gewährleisten, ist ein Ansatz die deduktive Überprüfung, bei der eine Sammlung von logischen Bedingungen erstellt wird, die das System erfüllen muss. Diese Bedingungen können mit automatischen Theorembeweisern wie SAT- und SMT-Solvern algorithmisch verifiziert werden. Ein anderer beliebter Ansatz ist die Modellprüfung, die darin besteht, das System durch mathematische Modelle darzustellen und bestimmte Eigenschaften, die oft in temporalen Logiken beschrieben werden, erschöpfend zu | | | | | |

prüfen. In diesem Kurs werden verschiedene Aspekte der formalen Verifikation und der Modellprüfung behandelt, darunter:

- Erfüllbarkeit von Sätzen: Auflösung und die wichtigsten Bestandteile moderner Erfüllbarkeitsbeweiser.
- Satisfiability modulo theories, insbesondere unter Verwendung linearer Ungleichungen, und der zugrunde liegende Simplex-Algorithmus.
Explizite Zustands- und symbolische Algorithmen für die Modellprüfung von temporalen Logiken mit linearer Zeit (LTL) und Verzweigungszeit (CTL) für endliche Maschinen.
- Symbolische Modellprüfung unter Verwendung von BDDs.
- Modellprüfung unter Wahrscheinlichkeiten und Ungewissheit, unter Verwendung von Markov-Ketten und Markov-Entscheidungsprozessen, mit der PCTL-Logik und den Eigenschaften der erwarteten Belohnung
- Die Technik des maschinellen Lernens (Reinforcement Learning) sowie Techniken der Modellprüfung und der formalen Verifikation, um ihre Anwendung auf kritischen Systemen sicher zu machen

Lehrformen

- 90 Minuten wöchentliche Vorlesung, vor Ort.
- 90-minütige Übungsstunde, vor Ort.
- Wöchentliche Übungen, sowohl praktisch als auch theoretisch

Prüfungsformen

50% Klausur (90 Minuten) + 50% Projektarbeit

Voraussetzungen für die Vergabe von Credits

Projekt und Modulabschlussprüfung bestanden.

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/97: M.Sc. Computer Science

5/105: M.Sc. Angewandte Informatik

5/91: M.Sc. IT-Sicherheit/Informationstechnik

Titel des Moduls: Foundations of Programming Languages, Verification, and Security
Foundations of Programming Languages, Verification, and Security

| | | | | | |
|---|------------------------|--------------------------|--|---------------------------------|---------------------------------------|
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester see examination regulations | Turnus Wintersemester | Dauer 1 Semester |
| Lehrveranstaltungen Foundations of Programming Languages, Verification, and Security (211062) | | | Kontaktzeit 4 SWS (60 h) | Selbststudium 90 h | Gruppengröße 20 Studierende |
| Unterrichtssprache Englisch | | | Teilnahmevoraussetzungen | | |

Modulbeauftragte/r und hauptamtlich Lehrende

Modulbeauftragte/r: Apl.-Prof. Dr. Catalin Hritcu
 Lehrende: Apl.-Prof. Dr. Catalin Hritcu

Verwendung des Moduls

M.Sc. Computer Science

M.Sc IT-Sicherheit/Netze und Systeme (Wahl oder Wahlpflicht)

M.Sc. IT-Sicherheit/Informationstechnik (Wahl oder Wahlpflicht)

M.Sc. Mathematik (Nebenfach Informatik)

Vorkenntnisse

This advanced course for MSc and PhD students requires having attended the Proofs are Programs course or at least having a working knowledge of the contents of the Logical Foundations book, including familiarity with logic, mechanized proofs, and functional programming in the Coq proof assistant.

Lernziele (learning outcomes)

After successful completion of this course, students will be able to

- understand how to define in Coq the syntax of simple programming languages, in particular variants of a simple imperative language and of the simply-typed lambda calculus;
- define the big-step and small-step operational semantics of such simple languages;
- formally define type systems for such languages as inductive relations;
- work out the metatheory of such languages, by proving results such as type soundness;
- understand the semantic foundations of Hoare Logic and Relational Hoare Logic;
- use Hoare Logic for verifying the correctness of simple imperative programs, both formally in Coq and informally on paper;
- understand the semantic foundations of Secure Information Flow Control and Noninterference.
- use Relational Hoare Logic for proving program equivalence as well as noninterference of simple imperative programs;
- be familiar with static and dynamic enforcement mechanisms for Secure Information Flow Control as well as their formal noninterference guarantees (e.g. security type systems, secure multi-execution, etc.);
- understand how to formalize Cryptographic Constant Time, Speculative Constant Time, and Speculative Load Hardening (SLH) in Coq;
- apply various proof techniques both in Coq and in informal paper proofs (e.g. induction on rule derivations) or just in Coq (e.g. proof automation).

Inhalt

Complex proofs on paper are difficult to construct, check, and maintain. This holds not only for interesting proofs in mathematics, but also for complex formal proofs about interesting programs. For this reason, machine-checked proofs created with the help of interactive tools called proof assistants are gaining increased traction in academia and industry. Proof assistants have been used to prove the correctness and security of realistic compilers, operating systems, cryptographic libraries, or smart contracts, and also to construct machine-checked proofs for challenging theorems in mathematics.

This course will use the Coq proof assistant [2] to lay down the foundations of Programming Languages, Verification, and Security. The Coq proof assistant enables us to program formal proofs interactively and it machine-checks the correctness of the proofs along the way. We will use Coq to define the syntax and semantics of imperative and functional programming languages, to define type systems, and to prove theorems such as type soundness. We will also formalize Hoare Logic and Relational Hoare Logic in Coq and use them to prove the correctness and security of simple imperative programs. Finally, the course will introduce static and dynamic enforcement mechanisms for Secure Information Flow Control as well as their formal noninterference guarantees. Finally we will formalize Cryptographic Constant Time and Speculative Constant Time and prove in Coq that a software defense called Speculative Load Hardening (SLH) achieves Speculative Constant Time.

This hands-on course is based on the Programming Languages Foundations online textbook [1], which is itself formalized and machine-checked in the Coq proof assistant. The many exercises in each book chapter are to be solved weekly mostly in Coq, from easy exercises allowing the students to practice concepts from the lecture, building incrementally to slightly more interesting programs and proofs and also to various optional challenges.

Lehrformen

This course consists of lectures and weekly exercises, in which the students will solve problems using the Coq proof assistant for which they can get help from a tutor.

Prüfungsformen

Written final exam (mandatory, 120 minutes) and exercise sheets.

Voraussetzungen für die Vergabe von Credits

There will be a mandatory written final exam (120 minutes) that counts for 60% of the grade and weekly exercise sheets that have to be submitted on time and that count for 40% of the grade. We will also have an optional midterm exam that helps students practice for the final exam, but only counts for bonus points, up to 10% of the final grade. One can additionally get bonus points up to 5% of the final grade by solving all exercise sheets.

To pass the course and receive credit points one has to attend the final exam and the weighed sum of your scores including bonus points (which can add up to a maximum of 115%) has to be at least 50%.

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/97: M.Sc. Computer Science

5/91: M.Sc. IT-Sicherheit/Informationstechnik [PO22]

5/99: M.Sc. IT-Sicherheit/Netze und Systeme [PO22]

| | | | | | |
|--|------------------------|--------------------------|---|---------------------------------|---------------------------------------|
| Titel des Moduls: Functional Programming Functional Programming | | | | | |
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung / see examination regulations | Turnus Wintersemester | Dauer 1 Semester |
| Lehrveranstaltungen Functional Programming (211060) | | | Kontaktzeit 4 SWS (60 h) | Selbststudium 90 h | Gruppengröße 50 Studierende |
| Unterrichtssprache Englisch | | | Teilnahmevoraussetzungen kein | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Dr. Clara Schneidewind Lehrende: Dr. Clara Schneidewind Dr. Jana Hofmann Dr. Catalin Hritcu | | | | | |
| Verwendung des Moduls B.Sc. Informatik B.Sc. Angewandte Informatik M.Sc. IT-Sicherheit / Informationstechnik | | | | | |
| Vorkenntnisse Es sind keine besonderen Vorkenntnisse erforderlich. | | | | | |
| Lernziele (learning outcomes) After successful completion of this course, students will: <ul style="list-style-type: none"> • develop programs in high-level, functional programming languages, in particular OCaml • understand and apply the use of recursion to define data structures (lists, maps, trees, etc.) and purely functional algorithms • understand the structure and advantages of type systems and use them to support program design and implementation • study advanced functional programming features, including type polymorphism and higher-order functions • reason informally about the correctness and efficiency of functional programs and be aware of more formal alternatives to reasoning • apply type abstraction and modularization to structure programs into collections of libraries and use those to build more complex programs on top of them • argue about the correctness and security of functional programs • understand the fundamental principles of programming language design, especially applied to functional programming • design and develop simple programming languages, covering their formal definition and subsequent implementation as interpreters | | | | | |
| Inhalt This course offers a rigorous and hands-on introduction to the principles and practice of functional programming—an increasingly influential paradigm that underpins the development of reliable, maintainable, and secure software. Functional programming centers on the use of pure functions: self-contained computations that produce outputs without altering external state. This absence of side effects leads to code that is not only elegant and expressive but also inherently more predictable and easier to reason about than traditional imperative code. As a result, functional programming is particularly well-suited for building security-critical systems, where correctness and | | | | | |

robustness are paramount.

The course uses OCaml, a modern functional language, to explore the theory and practice of functional programming from the ground up. Students will learn how strong static typing and advanced type systems serve as powerful tools for writing safe, composable, and error-resistant programs.

Through a mix of lectures, interactive exercises, and extended case studies (including the design and implementation of a small programming language), students will gain a deep understanding of how functional languages work and why they matter. These skills translate seamlessly to many modern mainstream languages, which increasingly incorporate functional concepts.

In addition to mastering practical programming techniques, students will also learn to reason formally about program correctness—an essential competency for those aiming to build high-assurance systems. By the end of the course, students will be equipped not just to write functional programs, but also to argue about their security and correctness.

Lehrformen

Vorlesung mit Übung

Prüfungsformen

Schriftliche Modulabschlussprüfung (120 Minuten)

Voraussetzungen für die Vergabe von Credits

Bestehen der Modulabschlussprüfung und erfolgreiche Teilnahme an den Übungen.

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/158: B.Sc. Informatik [PO 22]

5/170: B.Sc. Informatik [PO 20]

5/168: B.Sc. Angewandte Informatik [PO 22]

5/170: B.Sc. Angewandte Informatik [PO 20]

5/91: M.Sc. IT-Sicherheit / Informationstechnik [PO 22]

5/84: M.Sc. IT-Sicherheit / Informationstechnik [PO 20]

| | | | | | |
|---|------------------------|--------------------------|---|---------------------------------|------------------------------------|
| Titel des Moduls: Fundamentals of Data Science Fundamentals of Data Science | | | | | |
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung | Turnus Sommersemester | Dauer 1 Semester |
| Lehrveranstaltungen Fundamentals of Data Science (141213) | | | Kontaktzeit 60 h | Selbststudium 90 h | Gruppengröße Studierende |
| Unterrichtssprache Deutsch | | | Teilnahmevoraussetzungen | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr.-Ing. Aydin Sezgin Lehrende: Prof. Dr.-Ing. Aydin Sezgin | | | | | |
| Verwendung des Moduls M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. Informatik | | | | | |
| Vorkenntnisse Mathematik I-IV Systemtheorie I-III Optimierung | | | | | |
| Lernziele (learning outcomes) | | | | | |
| Inhalt Die Modulnote setzt sich aus zwei Anteilen zusammen: 1. Note der mündlichen Prüfung (36 %) 2. Note der Ausarbeitung eines wissenschaftlichen Artikels und des dazugehörigen Vortrags (64 %) Ausarbeitung: Für die Ausarbeitung sollte eine LaTeX-Vorlage (z.B. IEEEtran mit DIN A4, zweispaltiger Text) benutzt werden und 2 Seiten nicht überschreiten. Vortrag: Die Dauer des Vortrags ist 20 Minuten mit einer anschließenden Fragen- und Diskussionsrunde von 5-10 Minuten. Es ist empfehlenswert, den Vortrag allgemein verständlich zu halten. Backup-Folien werden empfohlen. Sprache: Der Vortrag kann wahlweise in Deutsch oder Englisch sein. | | | | | |
| Lehrformen Vorlesung mit Übung | | | | | |
| Prüfungsformen mündlich (30 min), Anmeldung: FlexNow Termin und Raum nach Absprache mit dem Dozenten | | | | | |
| Voraussetzungen für die Vergabe von Credits Bestandene Modulabschlussprüfung | | | | | |
| Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 5/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO22] M.Sc. Informatik [PO23] | | | | | |

| | | | | | |
|---|------------------------|--------------------------|---|---------------------------------|------------------------------------|
| Titel des Moduls: Fundamentals of GPU Programming Fundamentals of GPU Programming | | | | | |
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung | Turnus Wintersemester | Dauer 1 Semester |
| Lehrveranstaltungen Fundamentals of GPU Programming (141374) | | | Kontaktzeit 45 h | Selbststudium 105 h | Gruppengröße Studierende |
| Unterrichtssprache Englisch | | | Teilnahmevoraussetzungen | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Dr. Denis Eremin Lehrende: Dr. Denis Eremin | | | | | |
| Verwendung des Moduls M.Sc. Angewandte Informatik M.Sc. IT-Sicherheit/ Informationstechnik | | | | | |
| Vorkenntnisse C (Programmiersprache) | | | | | |
| Lernziele (learning outcomes) Die Studierenden erlernen das Programmieren auf Grafikprozessoren (GPUs) | | | | | |
| Inhalt Zu einem bestimmten Zeitpunkt um 2003 stieg die Rechenleistung nicht auf Kosten der Taktfrequenz des Prozessors, sondern durch Erhöhung der Anzahl der auf dem Prozessorchip zugewiesenen Rechenkern. Grafikprozessoren (GPUs) sind die Meister dieser Computer-Hardware-Entwicklung und bieten bis zu Zehntausende einzelner Kerneinheiten. Gleichzeitig wird das GPU-Speichersystem nicht so sehr durch die Kompatibilitätsanforderungen mit älteren Generationen eingeschränkt wie CPU-Speichersysteme. Deswegen zeigen GPUs im Vergleich zu ihren älteren "Bruder" -Zentraleinheiten (CPUs) eine deutlich bessere Rohleistung der Recheneinheiten und des Speichersystems. Ursprünglich für Videobearbeitungsaufgaben entwickelt, wird die enorme Rechenleistung moderner GPUs üblicherweise zur Unterstützung von CPUs oder zur Lösung einer Vielzahl von Rechenproblemen mit (massiv) parallelisierbaren Teilen verwendet, wodurch Teraflops-hohe Rechenleistung kann schon auf Laptop- / Desktop-Computers erzielt werden. Der vorliegende Kurs zeigt, wie CUDA C (Erweiterung der C-Sprache für die GPU-Programmierung) und das entsprechende (sehr flexible!) CUDA-Laufzeit-API-Framework verwendet werden kann, um die Ausführung einiger typischer Programmiermuster um einen Faktor von 10 oder mehr zu beschleunigen das der CPU. Ausgehend vom CUDA-Programmiermodell geht man zum CUDA-Ausführungsmodell über und betrachtet grundlegende konzeptionelle, Software- und Hardwareprobleme, die zum Verständnis der Funktionsweise von GPUs beitragen. Fallstudien zu mehreren Problemen mit massiv parallelen Algorithmen, die in GPUs implementiert sind, werden ebenfalls weiter ausgeführt. Das theoretische Wissen, das in den Vorlesungen vermittelt wird, wird durch eine Vielzahl von praktischen Beispielen untermauert, an denen die SchülerInnen zu Hause arbeiten können. | | | | | |
| Lehrformen Vorlesung mit Übung | | | | | |
| Prüfungsformen Hausarbeit | | | | | |
| Voraussetzungen für die Vergabe von Credits Bestehen der Hausarbeit | | | | | |

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/105: M.Sc. Angewandte Informatik

5/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

5/84: M.Sc. IT-Sicherheit/ Informationstechnik [PO 20]

Titel des Moduls: Highlights of Theoretical Computer Science [M.Sc]
Highlights of Theoretical Computer Science

| | | | | | |
|--|------------------------|--------------------------|--|---------------------------------|---------------------------------------|
| Modul-Nr./Code | Credits 9 CP | Workload 270 h | Semester see examination regulations/ siehe Prüfungsordnung | Turnus Sommersemester | Dauer 1 Semester |
| Lehrveranstaltungen Highlights of Theoretical Computer Science (211057) | | | Kontaktzeit 6 SWS (90 h) | Selbststudium | Gruppengröße 30 Studierende |
| Unterrichtssprache Englisch | | | Teilnahmevoraussetzungen Successful completion of an introductory course on theoretical computer science (covering formal languages, basics of complexity theory including NP-completeness and reductions, basics of computability theory). Interest and motivation to learn about theoretical concepts. | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Michael Walter Prof. Dr. Thomas Zeume Lehrende: Prof. Dr. Michael Walter Prof. Dr. Thomas Zeume Dr. Vladimir Lysikov | | | | | |
| Verwendung des Moduls M.Sc. Computer Science M.Sc. Angewandte Informatik M.Sc. IT-Sicherheit / Informationstechnik M.Sc. IT-Sicherheit / Netze und Systeme | | | | | |
| Vorkenntnisse | | | | | |
| Lernziele (learning outcomes) You will know some of the most important results and insights of modern theoretical computer science. You will learn approaches and techniques that go well beyond a first course. You will be able to recognize when these can be used and how to adapt them to new situations. You will be able to independently acquire new knowledge in this area. | | | | | |
| Inhalt The insights and techniques of modern theoretical computer science have been key for advances in all areas of computer science. In this course, we will discuss some highlights and the techniques that underpin them. Possible topics that we might cover: <ul style="list-style-type: none"> • Computational models (is there life beyond Turing machines?) • Kolmogorov complexity (what is the shortest program that produces some output?) • Communication complexity (how many bits must Alice and Bob exchange to jointly solve a problem?) • Fine-grained complexity (are some easy problems easier than others? and why?) • Fast multiplication of numbers and matrices (can you beat the high-school method?) • Randomness (does it really help to compute faster?) • Circuit lower bounds (why is it so hard to prove that problems are hard?) • Convex optimization (how to maximize profit if all you can ask are yes/no questions) | | | | | |

- Hardness of approximation (how easy is it to find near-optimal solutions?)
- Cryptography and computation

If you enjoyed your first course in theoretical computer science in the Bachelor's and would like to deepen your knowledge by getting an overview of the fascinating theory of computing, then this course will be exactly right for you.

Lehrformen

Vorlesung mit Übung

Prüfungsformen

Final module examination. Format will depend on number of participants.

Voraussetzungen für die Vergabe von Credits

Bestandene Modulabschlussprüfung

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

9/97: M.Sc. Computer Science

9/105: M.Sc. Angewandte Informatik

9/91: M.Sc. IT-Sicherheit / Informationstechnik [PO 22]

9/84: M.Sc. IT-Sicherheit / Informationstechnik [PO 20]

9/99: M.Sc. IT-Sicherheit / Netze und Systeme [PO 22]

9/96: M.Sc. IT-Sicherheit / Netze und Systeme [PO 20]

| | | | | | |
|---|------------------------|--------------------------|---|---------------------------------|---------------------------------------|
| Titel des Moduls: Human Aspects of Cryptography Adoption Human Aspects of Cryptography Adoption | | | | | |
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung | Turnus Wintersemester | Dauer 1 Semester |
| Lehrveranstaltungen Human Aspects of Cryptography Adoption | | | Kontaktzeit 60 h | Selbststudium 90 h | Gruppengröße 30 Studierende |
| Unterrichtssprache Englisch | | | Teilnahmevoraussetzungen Keine | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Martina Angela Sasse Lehrende: Prof. Dr. Martina Angela Sasse | | | | | |
| Verwendung des Moduls Master IT-Sicherheit/ Informationstechnik Master IT-Sicherheit/ Netze und Systeme | | | | | |
| Vorkenntnisse Keine | | | | | |
| Lernziele (learning outcomes) The aim of the lecture is to examine the reasons why <ol style="list-style-type: none"> 1. cryptographic solutions – which experts agree offer good protection against most of the common attacks today – are not adopted by most individuals and organisations, and 2. end-users, developers and system administrators who do use cryptographic solutions in some form frequently make mistakes that undermine the security protection. | | | | | |
| Inhalt In 1999, Whitten & Tygar's seminal USENIX paper "Why Johnny Can't Encrypt" established that people cannot use PGP encryption correctly, even with a graphical user interface and instruction. Over the past 20 years, there has been a string of Johnny papers on studies trying to encourage adoption or correct usage. The aim of this CASA lecture is to systematically examine the results of these studies and identify effective ways of promoting adoption and enable correct use of cryptography. <ul style="list-style-type: none"> • Usability, utility and technology adoption • Security threat models and people's mental models • Complexity or simplicity – who needs to know what? • Designing frictionless user journeys • Methods for testing and tweaking | | | | | |
| Lehrformen Vorlesung mit Übung | | | | | |
| Prüfungsformen Mündliche Prüfung | | | | | |
| Voraussetzungen für die Vergabe von Credits Bestandene Modulabschlussprüfung | | | | | |
| Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 5/91: M.Sc. IT-Sicherheit/ Informationstechnik | | | | | |

| | | | | | |
|---|------------------------|--------------------------|---|---------------------------------|---------------------------------------|
| Titel des Moduls: Information Theory Information Theory | | | | | |
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung / see examination regulations | Turnus Sommersemester | Dauer 1 Semester |
| Lehrveranstaltungen Information Theory (211007) | | | Kontaktzeit 60 h | Selbststudium 90 h | Gruppengröße 30 Studierende |
| Unterrichtssprache Englisch | | | Teilnahmevoraussetzungen Keine | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Michael Walter Lehrende: Prof. Dr. Michael Walter | | | | | |
| Verwendung des Moduls B.Sc. Informatik (bis SS 23) B.Sc. IT-Sicherheit M.Sc. Informatik M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Netze und Systeme (bis SS 23) M.Sc. Angewandte Informatik | | | | | |
| Vorkenntnisse Vertrautheit mit der diskreten Wahrscheinlichkeitsrechnung (wir werden Sie kurz an die wichtigsten Fakten erinnern). Einige Erfahrung mit präzisen mathematischen Aussagen und strengen Beweisen (da wir viele davon im Kurs sehen werden). Ein Teil der Hausaufgaben wird die Programmierung in Python erfordern. | | | | | |
| Lernziele (learning outcomes) Sie werden grundlegende Konzepte, Algorithmen und Ergebnisse der Informationstheorie kennenlernen. Nach erfolgreichem Abschluss dieses Kurses kennen Sie das mathematische Modell der Informationstheorie, wissen, wie man Algorithmen für eine Vielzahl von Informationsverarbeitungsaufgaben entwirft und analysiert, und wie man sie in Python implementiert. Sie haben sich selbstständig in ein Thema der Informationstheorie eingelesen und dieses vor Ihren Kommilitonen präsentiert. Sie werden auf einen weiterführenden Kurs oder ein Forschungs- oder Abschlussprojekt in diesem Bereich vorbereitet. Eine genaue Auflistung der Lernziele finden Sie auf der Homepage des Kurses. | | | | | |
| Inhalt Dieser Kurs gibt eine Einführung in die Informationstheorie - die mathematische Theorie der Information. Seit ihren Anfängen hat die Informationstheorie einen tiefgreifenden Einfluss auf die Gesellschaft gehabt. Sie bildet die Grundlage für wichtige technologische Entwicklungen, von zuverlässigen Speichern bis hin zu Mobilfunkstandards, und ihr vielseitiges mathematisches Instrumentarium findet Anwendung in der Informatik, dem maschinellen Lernen, der Physik, der Elektrotechnik, der Mathematik und vielen anderen Disziplinen. Ausgehend von der Wahrscheinlichkeitstheorie werden wir erörtern, wie man Informationsquellen und Kommunikationskanäle mathematisch modelliert, wie man Informationen optimal komprimiert und wie man fehlerkorrigierende Codes entwirft, die uns eine zuverlässige Kommunikation über verrauschte Kommunikationskanäle ermöglichen. Wir werden auch sehen, wie die in der Informationstheorie verwendeten | | | | | |

Techniken allgemeiner angewendet werden können, um Vorhersagen aus verrauschten Daten zu treffen.

Vorläufiger Lehrplan:

- Begrüßung, Einführung in die Informationstheorie
- Auffrischung der Wahrscheinlichkeitstheorie
- Numerische Zufallsvariablen, Konvexität und Konkavität, Entropie
- Symbol-Codes: Verlustfreie Komprimierung, Huffman-Algorithmus
- Block-Codes: Shannons Quellencodierungstheorem, sein Beweis und Variationen
- Strom-Codes: Lempel-Ziv-Algorithmus
- Strom-Codes: Arithmetische Kodierung
- Gemeinsame Entropien & Kommunikation über verrauschte Kanäle
- Shannons Theorem der verrauschten Kodierung
- Beweis des Theorems der verrauschten Kodierung (Noisy Coding Theorem)
- Beweis der Umkehrung, Shannons Theorie und Praxis
- Reed-Solomon-Codes
- Nachrichtenübermittlung für Dekodierung und Inferenz, Ausblick
- Studentische Präsentationen

Weitere Informationen finden Sie auf der Kurs-Homepage https://qi.rub.de/it_ss23.

Lehrformen

Vorlesung mit Übung

Prüfungsformen

Schriftliche (180 Minuten) oder mündliche (30 Minuten) Modulabschlussprüfung, abhängig von der Teilnehmerzahl. Wird zum Kursbeginn bekanntgegeben.

Voraussetzungen für die Vergabe von Credits

Passed Exam

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/158: B.Sc. Informatik [PO 22]

5/170: B.Sc. Informatik [PO 20]

5/150: B.Sc. IT-Sicherheit [PO 22]

5/149: B.Sc. IT-Sicherheit [PO 20]

5/97: M.Sc. Informatik

5/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

5/84: M.Sc. IT-Sicherheit/ Informationstechnik [PO 20]

5/99: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 22]

5/105: M.Sc. Angewandte Informatik

| Titel des Moduls: Introduction to Cybercrime and Incident Response | | | | | |
|--|----------------|-----------------|---|----------------------|---------------------|
| Introduction to Cybercrime and Incident Response | | | | | |
| Modul-Nr./Code | Credits | Workload | Semester | Turnus | Dauer |
| | 4 CP | 120h | siehe Prüfungsordnung / see examination regulations | Wintersemester | 1 Semester |
| Lehrveranstaltungen | | | Kontaktzeit | Selbststudium | Gruppengröße |
| Introduction to Cybercrime and Incident Response (212042) | | | 40h (eine Woche im Block) (3 SWS) | 80h | 30 Studierende |
| Unterrichtssprache | | | Teilnahmevoraussetzungen | | |
| Englisch/Deutsch | | | | | |
| Modulbeauftragte/r und hauptamtlich Lehrende | | | | | |
| Modulbeauftragte/r: M.Sc. Burak Uslu (Lehrbeauftragter) Lehrende: M.Sc. Burak Uslu | | | | | |
| Verwendung des Moduls | | | | | |
| M.Sc. IT-Sicherheit/Informationstechnik M.Sc. IT-Sicherheit/Netze und Systeme | | | | | |
| Vorkenntnisse | | | | | |
| Grundkenntnisse über Schadsoftware | | | | | |
| Lernziele (learning outcomes) | | | | | |
| <ul style="list-style-type: none"> Nach erfolgreicher Teilnahme an der Vorlesung kennen die Studierenden den aktuellen Stand der Cyberkriminalität und der Incident Response Praxis in Deutschland. Der Kurs behandelt die Analyse von Vorfällen, die Prävention von Folgeangriffen und Methoden zur Verbesserung von Systemen, um sie gegen zukünftige Vorfälle zu härten. | | | | | |
| Inhalt | | | | | |
| Cyberkriminalität ist ein reales Beispiel für die Ausnutzung von Software und Systemen mit dem Ziel, persönliche Vorteile zu erlangen. Incident-Response-Maßnahmen versuchen, gegen Cyberkriminalität vorzugehen und umfassen verschiedene Techniken, die zur Identifizierung von Akteuren und den Folgen krimineller Aktivitäten eingesetzt werden können. In diesem Kurs lernen wir den aktuellen Stand der Internetkriminalität in Deutschland kennen. Dabei betrachten wir die Entwicklung der Cyberkriminalität in den letzten Jahren und welche Auswirkungen sie auf die Gesellschaft hat. Darüber hinaus untersuchen wir reale Vorfälle der Vergangenheit, betrachten verschiedene Möglichkeiten der Prävention und lernen mehr über die derzeit existierenden Verfahren zur Reaktion auf Vorfälle. | | | | | |
| Lehrformen | | | | | |
| Vorlesung und praktische Aufgaben, wird als Blockkurs innerhalb von 1 Woche abgehalten. | | | | | |
| Prüfungsformen | | | | | |
| Schriftliche Prüfung | | | | | |
| Voraussetzungen für die Vergabe von Credits | | | | | |
| Bestehende Note in der Klausur. Als Vorleistung für die Teilnahme an der Klausur ist eine Gruppenpräsentation zu leisten. | | | | | |
| Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) | | | | | |
| 4/91: M.Sc. IT-Sicherheit/Informationstechnik 4/99: M.Sc. IT-Sicherheit/Netze und Systeme | | | | | |

Titel des Moduls: Komplexitätstheorie
Computational complexity theory

| | | | | | |
|--|------------------------|--------------------------|---|---------------------------------|------------------------------------|
| Modul-Nr./Code | Credits 9 CP | Workload 270 h | Semester siehe Prüfungsordnung / see examination regulations | Turnus Wintersemester | Dauer 1 Semester |
| Lehrveranstaltungen Computational complexity theory (211028) | | | Kontaktzeit 6 SWS (90 h) | Selbststudium 180 h | Gruppengröße Studierende |
| Unterrichtssprache Englisch | | | Teilnahmevoraussetzungen | | |

Modulbeauftragte/r und hauptamtlich Lehrende

Modulbeauftragte/r: Prof. Thomas Zeume
 Lehrende: Prof. Thomas Zeume

Verwendung des Moduls

M.Sc. Computer Science
 M.Sc. IT-Sicherheit/Informationstechnik
 M.Sc. IT-Sicherheit/Netze und Systeme
 M.Sc. Angewandte Informatik (nur bis SS 23)

Vorkenntnisse

Kenntnisse aus einem Grundkurs in theoretischer Informatik (Grundlagen der Komplexitätstheorie einschließlich NP-Vollständigkeit und Reduktionen) werden erwartet.

Lernziele (learning outcomes)

Die Studierenden lernen, algorithmische Probleme bezüglich ihrer Komplexität einzuordnen und so geeignete algorithmische Techniken zu ihrer Lösung zu identifizieren. Sie können insbesondere algorithmische Methoden für NP-vollständige Probleme anwenden. Sie können mit unterschiedlichen Berechnungsmodellen umgehen und sind in der Lage, einfache Aussagen über sie zu beweisen. Sie lernen im Diskurs eigene und fremde Lösungsansätze zu bewerten.

Inhalt

Die Komplexitätstheorie untersucht und klassifiziert Berechnungsprobleme bezüglich ihrer algorithmischen Schwierigkeit. Ziel ist es, den inhärenten Ressourcenverbrauch bezüglich verschiedener Ressourcen wie Rechenzeit oder Speicherplatz zu bestimmen, und Probleme mit ähnlichem Ressourcenverbrauch in Komplexitätsklassen zusammenzufassen. Die bekanntesten Komplexitätsklassen sind sicherlich P und NP, die die in polynomieller Zeit lösbaren bzw. verifizierbaren Probleme umfassen. Die Frage, ob P und NP verschieden sind, wird als eine der bedeutendsten offenen Fragen der theoretischen Informatik, ja sogar der Mathematik, angesehen. P und NP sind jedoch nur zwei Beispiele von Komplexitätsklassen. Andere Klassen ergeben sich unter anderem bei der Untersuchung der des benötigten Speicherplatzes, der effizienten Parallelisierbarkeit von Problemen, der Lösbarkeit durch zufallsgesteuerte Algorithmen, und der approximativen Lösbarkeit von Problemen. Die Vorlesung hat das Ziel, einen breiten Überblick über die grundlegenden Konzepte und Resultate der Komplexitätstheorie zu geben:

- Klassische Resultate für Platz- und Zeitkomplexitätsklassen: z.B. die Korrespondenz zwischen Spielen und Speicherplatz-Beschränkungen, der Nachweis, dass sich mit mehr Platz oder Zeit auch mehr Probleme lösen lassen, weitere grundlegende Beziehungen zwischen Zeit- und Platzbasierten Klassen, und die Komplexitätswelt zwischen NP und PSPACE
- Grundzüge der Komplexitätstheorie paralleler, zufallsbasierter und approximativer Algorithmen
- Einführung in ausgewählte neuere Themen: Komplexitätstheorie des interaktiven Rechnens, des probabilistischen Beweisens und Fine-grained Complexity.

Lehrformen

Vorlesung mit Übungen

Prüfungsformen

Mündliche Modulabschlussprüfung (20-30 Minuten)

Voraussetzungen für die Vergabe von Credits

Bestandene mündliche Modulabschlussprüfung

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

9/97: M.Sc. Computer Science

9/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

9/84: M.Sc. IT-Sicherheit/ Informationstechnik [PO 20]

9/99: M.Sc. IT-Sicherheit/ Netze und Systeme [PO22]

9/96: M.Sc. IT-Sicherheit/ Netze und Systeme [PO20]

| | | | | | |
|---|------------------------|--------------------------|---|---------------------------------|------------------------------------|
| Titel des Moduls: Kryptographische Protokolle Cryptographic protocols | | | | | |
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung / see examination regulations | Turnus Sommersemester | Dauer 1 Semester |
| Lehrveranstaltungen Kryptographische Protokolle (211031) | | | Kontaktzeit 4 SWS (60 h) | Selbststudium 90 h | Gruppengröße Studierende |
| Unterrichtssprache Englisch | | | Teilnahmevoraussetzungen | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Eike Kiltz Lehrende: Prof. Dr. Eike Kiltz | | | | | |
| Verwendung des Moduls M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Netze und Systeme M.Sc. Angewandte Informatik [bis SS 23] M.Sc. Computer Science | | | | | |
| Vorkenntnisse Inhalte des Moduls Kryptographie | | | | | |
| Lernziele (learning outcomes) <ul style="list-style-type: none"> • Vertiefung des Verständnisses für beweisbare Sicherheit • Schreiben von fehlerfreien Sicherheitsreduktionen • Neue Techniken für Sicherheitsbeweise • Erlernen fortgeschrittener kryptographischer Konstruktionen | | | | | |
| Inhalt Die Vorlesung beschäftigt sich mit erweiterten kryptographischen Protokollen und deren Anwendungen. Themenübersicht: <ul style="list-style-type: none"> • Game-based security definitions and proofs • Bilinear maps • Digital Signatures • Identification Protocols • Zero-Knowledge Proofs • Identity-based Encryption • CCA-secure encryption | | | | | |
| Lehrformen Vorlesung mit Übung | | | | | |
| Prüfungsformen Mündliche (30 Minuten) oder schriftliche Modulabschlussprüfung (120 Minuten), abhängig von der Teilnehmerzahl. Wird zu Beginn des Kurses mitgeteilt. | | | | | |
| Voraussetzungen für die Vergabe von Credits Bestandene Modulabschlussprüfung. | | | | | |

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

5/84: M.Sc. IT-Sicherheit/ Informationstechnik [PO 20]

5/99: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 22]

5/96: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 20]

5/97: M.Sc. Computer Science

Titel des Moduls: Machine Learning: Supervised Methods (kein Angebot im SS 25)
Machine Learning: Supervised Methods (no offer in SS 25)

| | | | | | |
|---|------------------------|--------------------------|---|---------------------------------|---------------------------------------|
| Modul-Nr./Code | Credits 6 CP | Workload 180 h | Semester siehe Prüfungsordnung / see examination regulations | Turnus Sommersemester | Dauer 1 Semester |
| Lehrveranstaltungen Machine Learning: Supervised Methods (211024) | | | Kontaktzeit 4 SWS (60 h) | Selbststudium 120 h | Gruppengröße 80 Studierende |
| Unterrichtssprache Englisch | | | Teilnahmevoraussetzungen | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Tobias Glasmachers Lehrende: Prof. Dr. Tobias Glasmachers | | | | | |
| Verwendung des Moduls M.Sc. Computer Science M.Sc. Angewandte Informatik M.Sc. IT-Sicherheit/ Informationstechnik | | | | | |
| Vorkenntnisse empfohlen: Vorlesung "Mathematics for Modeling and Data Analysis" | | | | | |
| Lernziele (learning outcomes) Internationalisierung: Die Veranstaltung wird auf Englisch durchgeführt. Digitalisierung: Inhalte werden durch Videos und Lesematerial vermittelt. Übungsaufgaben mit Programmieranteilen werden in Form von Jupyter-Notebooks bereitgestellt. Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • verstehen die Teilnehmer die Grundlagen der statistischen Lerntheorie • kennen die Teilnehmer die wichtigsten Algorithmen des überwachten statistischen Lernens und können diese auf Lernprobleme anwenden, • kennen die Teilnehmer Stärken und Beschränkungen verschiedenen Lernverfahren, • können die Teilnehmer Standardsoftware zum maschinellen Lernen zur Lösung neuer Probleme einsetzen. | | | | | |
| Inhalt Grundlagen der statistischen Lerntheorie, Querschnitt der wichtigsten Algorithmen des maschinellen Lernens, konkrete Problemlösung mit Standardsoftware. | | | | | |
| Lehrformen Vorlesung mit Übung im flipped classroom Format | | | | | |
| Prüfungsformen Schriftliche Modulabschlussprüfung (90 Minuten) | | | | | |
| Voraussetzungen für die Vergabe von Credits Das Bestehen des Kurses ist ein zweistufiger Prozess. Die erste Stufe ist ein aktiver Beitrag während des Semesters, dessen Einzelheiten in einer in einer der ersten Sitzungen bekannt gegeben werden. Die zweite Stufe ist eine schriftliche Prüfung von 90 Minuten. Der aktive Beitrag während des Semesters wird nicht benotet, ist aber Voraussetzung für die Teilnahme an der Prüfung. Die Note wird basiert ausschließlich auf der Abschlussprüfung. | | | | | |
| Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) | | | | | |

6/105: M.Sc. Angewandte Informatik

6/97: M.Sc. Computer Science

6/91: M.Sc IT-Sicherheit/ Informationstechnik [PO 22]

6/84: M.Sc IT-Sicherheit/ Informationstechnik [PO 20]

| Titel des Moduls: Master Praktikum/Projektarbeit IT-Sicherheit | | | | | |
|--|------------------------|--------------------------|--|---|------------------------------------|
| Modul-Nr./Code | Credits 4 CP | Workload 120 h | Semester 3 | Turnus jedes Semester | Dauer 1 Semester |
| Lehrveranstaltungen In jedem Semester wird eine wechselnde Auswahl an Praktika bereitgestellt. Die zugeordneten Veranstaltungen können im Vorlesungsverzeichnis eingesehen werden. | | | Kontaktzeit je nach Veranstaltungswahl | Selbststudium abhängig von der Praktikumswahl | Gruppengröße Studierende |
| Unterrichtssprache abhängig von der Praktikumswahl: Deutsch oder Englisch | | | Teilnahmevoraussetzungen | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Studiendekan Lehrende: siehe Praktikumsbeschreibung | | | | | |
| Verwendung des Moduls M.Sc. IT-Sicherheit / Informationstechnik M.Sc. IT-Sicherheit / Netze und Systeme | | | | | |
| Vorkenntnisse abhängig vom gewählten Praktikum | | | | | |
| Lernziele (learning outcomes) Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> haben Studierende Ihre praktischen Fähigkeiten in der Analyse und dem Einsatz von Verfahren zur Sicherung von IT-Systemen in einem Forschungs- oder Anwendungsbereich vertieft und erweitert je nach gewähltem Praktikum können noch weitere Lernziele dazu kommen | | | | | |
| Inhalt Es werden in jedem Semester einige Praktika und Projekt angeboten. Z.B.: Master-Praktikum Reverse-Engineering Security Features, Projekt Netz- und Datensicherheit, Forschungspraktikum Human-Centred Security. Die im Semester angebotenen Praktika sowie weiterführende Informationen zu den jeweiligen Praktika finden Sie im Vorlesungsverzeichnis im Modul "Master Praktikum/Projektarbeit IT-Sicherheit" unter "Veranstaltungen". | | | | | |
| Lehrformen Praktikum im Block oder als semesterbegleitende Veranstaltung | | | | | |
| Prüfungsformen Praktikum | | | | | |
| Voraussetzungen für die Vergabe von Credits | | | | | |
| Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) unbenotet | | | | | |

| | | | | | |
|--|------------------------|--------------------------|---|---------------------------------|------------------------------------|
| Titel des Moduls: Menschliches Verhalten in der IT-Sicherheit | | | | | |
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung | Turnus Sommersemester | Dauer 1 Semester |
| Lehrveranstaltungen Menschliches Verhalten in der IT-Sicherheit (211033) | | | Kontaktzeit 60 h | Selbststudium 90 h | Gruppengröße Studierende |
| Unterrichtssprache Deutsch | | | Teilnahmevoraussetzungen Keine | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Martina Angela Sasse Lehrende: Prof. Dr. Martina Angela Sasse M. Sc. Jonas Hielscher | | | | | |
| Verwendung des Moduls M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Netze und Systeme | | | | | |
| Vorkenntnisse Der vorherige Besuch der Vorlesung "Einführung in die Usable Security and Privacy" wird empfohlen | | | | | |
| Lernziele (learning outcomes) Die Veranstaltung vermittelt theoretische und praktische Kenntnisse über Forschungsmethoden im Bereich usable Security mit einem besonderen Schwerpunkt auf Laborstudien. Es werden theoretische Kenntnisse vermittelt, auf deren Grundlage die Studierenden selbstständig eine Laborstudie planen und umsetzen und auf diese Weise praktische Kenntnisse erwerben sollen. | | | | | |
| Inhalt In <i>Menschliches Verhalten in der IT-Sicherheit</i> lernt ihr, welche Faktoren Einfluss auf das Sicherheitsverhalten von Angestellten in Unternehmen und Nutzenden im Alltag nehmen, und welche Möglichkeiten bestehen, dieses zu beeinflussen und verändern. Außerdem wird vermittelt, warum bestehende Ansätze des Information Security Management (auch nach ISO 27000) in der Praxis oft nicht funktionieren und wie wir sie erweitern bzw. anpassen sollten. Studierende werden befähigt IT-Sicherheit in Organisationen aus einem ganzheitlichen Ansatz heraus zu betrachten, was unter anderem zwingend erforderlich ist um später Sicherheitsführungsaufgaben wahrzunehmen. Die Vorlesungsinhalte sind dabei umfangreich mit Erfahrungen aus der Praxis angereichert. | | | | | |
| Lehrformen Vorlesung und Übung | | | | | |
| Prüfungsformen Schriftliche Modulabschlussprüfung (120 Minuten) | | | | | |
| Voraussetzungen für die Vergabe von Credits Bestandene Modulabschlussprüfung | | | | | |
| Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 5/99: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 22] 5/96: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 20] 5/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22] | | | | | |

| | | | | | |
|---|------------------------|--------------------------|---|---------------------------------|------------------------------------|
| Titel des Moduls: Message Level Security Message Level Security | | | | | |
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung | Turnus Wintersemester | Dauer 1 Semester |
| Lehrveranstaltungen Message-Level Security (212060) | | | Kontaktzeit 60 h | Selbststudium 90 h | Gruppengröße Studierende |
| Unterrichtssprache Deutsch | | | Teilnahmevoraussetzungen | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Dr.-Ing. Christan Mainka Lehrende: Dr.-Ing. Christan Mainka Dr.-Ing. Vladislav Mladenov | | | | | |
| Verwendung des Moduls M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Netze und Systeme | | | | | |
| Vorkenntnisse Vorlesung Netzsicherheit 2, Grundkenntnisse der englischen Sprache, da diese die Sprache für Folien, Übungsaufgaben und die Virtuelle Maschine ist. | | | | | |
| Lernziele (learning outcomes) Studierende verfügen nach erfolgreichem Abschluss der Vorlesung über ein umfassendes Verständnis der Sicherheit der folgenden Technologien: Datenformate im Web, REST APIs, Authentifizierungs- und Autorisierungsprotokollen und Dokumentenformaten. Durch die praxisnahe Arbeit im Rahmen der Übungen bauen die Studierenden ihre Recherche-Fähigkeiten aus und erlernen weiterhin den sicheren Umgang mit verschiedenen Penetrationswerkzeugen. Am Ende der Vorlesung sind die Studierenden in der Lage, systematisch umfassende Sicherheitsanalysen sowie praktische Angriffe auf die behandelten Technologien selbstständig durchzuführen. Weiterhin sind die Studierenden in der Lage, das erlernte Wissen auf andere Technologien zu übertragen und komplexere Angriffsmöglichkeiten selbst durch kreatives Denken zu finden und auszunutzen. | | | | | |
| Inhalt Die Vorlesung behandelt das Thema Message-Level Security. Anders als bei SSL/TLS, welches einen sicheren Transportkanal aufbaut, geht es bei Message-Level Security darum, Nachrichten – wie HTTP Requests – auf Nachrichtenebene zu schützen. Hierbei kommt es auf die korrekte Verwendung von kryptografischen Verfahren als auch eine sichere Bereitstellung von API-Schnittstellen an. Im Rahmen der Vorlesung werden verschiedene Verfahren von Message-Level Security beleuchtet: | | | | | |
| <ul style="list-style-type: none"> • JSON ist eine universelle Datenbeschreibungssprache, die unter anderem von jedem modernen Browser unterstützt wird. Mithilfe von JSON-Signature und JSON-Encryption können JSON Nachrichten direkt geschützt werden. Doch reicht das aus oder können diese Sicherheitsmechanismen umgangen werden? • OAuth ist eine sehr weitverbreitete Technologie zum Delegieren von Berechtigungen und wird heutzutage von allen großen Webseiten wie Facebook, Google, Twitter, Github usw. eingesetzt. Die Vorlesung erklärt tiefgehende Details und gängige Fehler/Angriffe, die bei der Verwendung von OAuth entstehen können. • OpenID Connect ist eine Erweiterung für OAuth, um Benutzer:innen auf Webseiten mithilfe eines Drittanbieters zu authentifizieren (z. B. mittels Single Sign-On Verfahren wie „Sign in with Google“). OpenID Connect hat sich in den letzten Jahren zum de facto Standard für Web-Logins über Drittanbieter etabliert. In der Vorlesung wird detailliert erklärt, was die Unterschiede zu OAuth sind und welche Angriffe auf OpenID Connect möglich sind. In den praktischen Übungen können Sie Ihre Exploit-Fähigkeiten unter Beweis stellen. Schaffen wir es, den Account des Opfers übernehmen? • SAML steht für Security Assertion Markup Language und ist ein Single Sign-On Standard, der eine | | | | | |

weitgehende Verbreitung in Business-Szenerien findet. Allerdings existieren zahlreiche Angriffe von Identitätsdiebstahl bis hin zu Remote Code Execution.

- **PDF** ist das vermutlich am weitesten verbreitete universelle Dokumentenaustauschformat. In der Vorlesung werden die Sicherheitseigenschaften von PDFs beleuchtet. Insbesondere werden hierbei digitale Signaturen untersucht, welche z. B. bei Verträgen zum Einsatz kommen. Wird es uns gelingen, signierte Dokumente zu fälschen?

Den Studierenden wird ein tiefgehendes Verständnis der Systeme vermittelt. Zu allen untersuchten Systemen werden Angriffe vorgestellt, die sowohl aus der akademischen Welt als auch aus der Pentesting-Community stammen. Die Übungen bieten die Möglichkeit, das erlernte Wissen praktisch auszuprobieren. Hierzu erhalten die Studierenden eine virtuelle Maschine.

Lehrformen

Vorlesung mit Übung

Prüfungsformen

Schriftliche Modulabschlussprüfung (120 Minuten)

Voraussetzungen für die Vergabe von Credits

Bestandene schriftliche Modulabschlussprüfung

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

5/84: M.Sc. IT-Sicherheit/ Informationstechnik [PO 20]

5/99: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 22]

5/96: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 20]

Titel des Moduls: Microarchitectural Attacks and Defenses**Microarchitectural Attacks and Defenses**

| | | | | | |
|--|------------------------|--------------------------|---|---------------------------------|---------------------------------------|
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung / see examination regulations | Turnus Wintersemester | Dauer 1 Semester |
| Lehrveranstaltungen Microarchitectural Attacks and Defenses (212064) | | | Kontaktzeit 4 SWS (60 h) | Selbststudium 90 h | Gruppengröße 30 Studierende |
| Unterrichtssprache Englisch | | | Teilnahmevoraussetzungen keine | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Yuval Yarom Lehrende: Prof. Yuval Yarom | | | | | |
| Verwendung des Moduls M.Sc. ITS - Informationstechnik M.Sc. ITS - Netze und Systeme M.Sc. Computer Science | | | | | |
| Vorkenntnisse Der Kurs setzt voraus, dass die Teilnehmer in C programmieren können oder die Sprache im Laufe des Kurses erlernen. Sie brauchen genügend Erfahrung, um auf entfernten Rechnern unter Verwendung von SSH zu programmieren. Grundlegende Kenntnisse über die Funktionsweise von Computern, Assemblersprache und die Rolle des Betriebssystems sind erforderlich. Ein Verständnis grundlegender Konzepte der Computersicherheit (Sicherheitsbereiche, Schwachstellen usw.) und Vertrautheit mit grundlegender Kryptographie (AES, RSA, ECC) ist hilfreich. | | | | | |
| Lernziele (learning outcomes) <ul style="list-style-type: none">• Diagnose mikroarchitektonischer Schwachstellen• Bewertung der Widerstandsfähigkeit von Software gegen Schwachstellen in der Mikroarchitektur• Entwurf und Programmierung von Proof-of-Concept-Exploits für anfällige Software und Hardware• Entwurf und Implementierung von Gegenmaßnahmen für Software, die auf anfälliger Hardware ausgeführt wird | | | | | |
| Inhalt Der Kurs deckt den Bereich der Angriffe auf die Mikroarchitektur und deren Verteidigung ab. Er beginnt mit Cache-Angriffen und behandelt die wichtigsten Techniken (Prime+Probe, Evict+Time und Flush+Reload). Darauf aufbauend werden Varianten der Angriffe auf andere Speicherelemente sowie Angriffe, die Bandbreitenbeschränkungen ausnutzen, untersucht. Parallel zur Erforschung dieser Angriffe werden verschiedene Gegenmaßnahmen beschrieben, wobei der Schwerpunkt auf der Programmierung mit konstanter Zeit liegt. Der Kurs wechselt dann zu Angriffen auf spekulative Ausführung, wobei die verschiedenen Angriffe, Verteidigungsmaßnahmen und Gegenangriffe identifiziert und klassifiziert werden. Der Kurs behandelt außerdem verschiedene verwandte Angriffe, darunter Rowhammer und spannungs- und frequenzbasierte Angriffe. Darüber hinaus widmet der Kurs den Angriffsszenarien besondere Aufmerksamkeit, wobei insbesondere Angriffe auf den Betriebssystemkern, webbasierte und andere Remote-Angriffe sowie Angriffe auf vertrauenswürdige Ausführungsumgebungen untersucht werden. Ein besonderer Schwerpunkt des Kurses liegt auf der praktischen Umsetzung von Angriffs- und Abwehrtechniken. | | | | | |
| Lehrformen Vorlesung mit Übung | | | | | |

Prüfungsformen

Projektarbeiten mit Einreichung der Ergebnisse.

Voraussetzungen für die Vergabe von Credits

Bestandene Projektarbeiten mit schriftlichen Einreichungen.

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/91: M.Sc. ITS - Informationstechnik [PO 22]

5/84: M.Sc. ITS - Informationstechnik [PO 20]

5/99: M.Sc. ITS - Netze und Systeme [PO 22]

5/96: M.Sc. ITS - Netze und Systeme [PO 20]

5/97: M.Sc. Computer Science

| | | | | | |
|--|------------------------|--------------------------|---|---------------------------------|------------------------------------|
| Titel des Moduls: Mobile Network Security | | | | | |
| Mobile Network Security | | | | | |
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung / see examination regulations | Turnus Sommersemester | Dauer 1 Semester |
| Lehrveranstaltungen Mobile Network Security (211012) | | | Kontaktzeit 60h (4 SWS) | Selbststudium 90 h | Gruppengröße Studierende |
| Unterrichtssprache English | | | Teilnahmevoraussetzungen None | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Katharina Kohls Lehrende: Prof. Dr. Katharina Kohls | | | | | |
| Verwendung des Moduls | | | | | |
| Vorkenntnisse The lecture focuses on mobile network security. Prior knowledge in the context of computer networks and their security mechanisms helps to understand the technical concepts that will be addressed in the lecture. | | | | | |
| Lernziele (learning outcomes) <ul style="list-style-type: none"> • Knowing mobile network architectures and their components in 4G and 5G networks • Understanding existing attacks and their attack vectors, as well as security mechanisms that avoid known attacks • Experience with scientific work in mobile network security | | | | | |
| Inhalt Mobile networks are an integral part of our everyday lives. Their use cases range from casual web browsing over campus networks in industrial environments to first responder communication. In this course, we cover the technical aspects of mobile networks and address their security capabilities. After an introduction to the technical foundations of mobile network deployments, we will go into detail with scientific work on existing attacks against 4G and 5G networks. To this end, we analyze open attack vectors and discuss the consequences of attacks if being conducted in real-world infrastructures. | | | | | |
| Lehrformen The course consists of lectures that provide theoretical knowledge and practical exercises that help to apply the contents of the lectures. | | | | | |
| Prüfungsformen Written exam with a duration of 120 minutes. | | | | | |
| Voraussetzungen für die Vergabe von Credits Passing the exam | | | | | |
| Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 5/97: M.Sc. Computer Science 5/105: M.Sc. Angewandte Informatik 5/91: M.Sc. IT-Sicherheit / Informationstechnik 5/99: M.Sc. IT-Sicherheit/ Netze und Systeme | | | | | |

Titel des Moduls: Physical Attacks and Countermeasures
Physical Attacks and Countermeasures

| | | | | | |
|-----------------------|------------------------|--------------------------|---|---------------------------------|----------------------------|
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung / see examination regulations | Turnus Sommersemester | Dauer 1 Semester |
|-----------------------|------------------------|--------------------------|---|---------------------------------|----------------------------|

| | | | |
|---|-----------------------------------|------------------------------|------------------------------------|
| Lehrveranstaltungen Physical Attacks and Countermeasures (211034) | Kontaktzeit 4SWS (60 h) | Selbststudium 90 h | Gruppengröße Studierende |
|---|-----------------------------------|------------------------------|------------------------------------|

| | |
|---------------------------------------|---------------------------------|
| Unterrichtssprache Englisch | Teilnahmevoraussetzungen |
|---------------------------------------|---------------------------------|

Modulbeauftragte/r und hauptamtlich Lehrende
 Modulbeauftragte/r: Dr. Jan Richter-Brockmann
 Lehrende: Dr. Jan Richter-Brockmann

Verwendung des Moduls
 M.Sc. IT-Sicherheit/ Informationstechnik
 M.Sc. IT-Sicherheit/ Netze und Systeme
 M.Sc. Computer Science

Vorkenntnisse
 Verständnis der englischen Sprache, Grundkenntnisse der Digitaltechnik, Grundkenntnisse der Datensicherheit und Kryptographie, solide Programmierkenntnisse in mindestens einer Programmiersprache (z.B. C++), Grundkenntnisse der Computerarchitektur, Grundkenntnisse der Signalverarbeitung.

Lernziele (learning outcomes)
 Die Studierenden

- verstehen wie und warum physikalische Angriffe funktionieren.
- sind in der Lage Messdaten anhand der erlernten Methoden auszuwerten und die Sicherheit einer Implementierung zu bewerten.
- erkennen die Gefahr von physikalischen Angriffen für Implementierungen von kryptographischen Algorithmen.
- kennen mögliche Gegenmaßnahmen und wissen, wie diese anzuwenden sind, um ein System gegen physikalische Angriffe zu schützen.

Inhalt

Moderne kryptographische Algorithmen bieten ausreichend Schutz gegen die bekannten mathematischen und kryptanalytischen Angriffe. In der Praxis werden diese Algorithmen für sicherheitskritische Anwendungen auf verschiedenen Plattformen implementiert. Dies geschieht sowohl als Programmcode (Software) als auch mit logischen Elementen/Schaltungen (Hardware). Der physikalische Zugang zu kryptographischen Implementierungen (z.B., eine Smartcard oder ein Smartphone, welche zum Bezahlen benutzt werden), in welchen der geheime Schlüssel eingebettet ist, hat zur Entstehung einer neuen Klasse von Angriffen, genannt physikalische Angriffe, geführt. Diese Angriffe zielen darauf ab den geheimen Schlüssel, welcher vom kryptographischen Algorithmus benutzt wird, zu extrahieren. Ein erfolgreicher physikalischer Angriff deutet nicht auf Schwächen im Algorithmus sondern auf Schwachstellen in der Implementierung hin. Daher müssen bereits in der Entwicklungsphase von kryptographischen Implementierungen physikalische Angriffe als potenzielles Risiko berücksichtigt und bestmöglich verhindert werden.

Das Ziel dieser Lehrveranstaltung ist es einen Überblick über bekannte physikalische Angriffe und deren

Gegenmaßnahmen zu geben. Im ersten Teil der Vorlesung werden die verschiedenen Angriffstypen eingeführt, während im zweiten Teil der Fokus auf Gegenmaßnahmen liegt.

Lehrformen

Vorlesung mit Übung

Prüfungsformen

Schriftliche Prüfung (120 Minuten) und Projektarbeit (semesterbegleitend)

Voraussetzungen für die Vergabe von Credits

Projektbasiertes Arbeiten ist ein großer Teil der Lehrveranstaltung. Zusätzlich zu einer schriftlichen Prüfung gibt es wöchentliche Projektarbeiten (Hausaufgaben). Beide Teile müssen individuell bearbeitet werden, sind bewertet und gehen in die Endnote ein. Dabei werden die beiden Teile wie folgt bewertet:

Wöchentliche Projektarbeiten (Hausaufgaben): 30

Klausur: 70

Der Erwerb von Bonuspunkten für die Klausur ist möglich.

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

5/99: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 22]

5/97: M.Sc. Computer Science

Titel des Moduls: Privacy Engineering, Data Governance and Usability
Privacy Engineering, Data Governance and Usability

| | | | | | |
|---|------------------------|--------------------------|---|---------------------------------|---------------------------------------|
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung | Turnus Wintersemester | Dauer 1 Semester |
| Lehrveranstaltungen Privacy Engineering, data governace and usability (212037) | | | Kontaktzeit 45 h | Selbststudium 105 h | Gruppengröße 20 Studierende |
| Unterrichtssprache Englisch | | | Teilnahmevoraussetzungen | | |

Modulbeauftragte/r und hauptamtlich Lehrende

Modulbeauftragte/r: Dr. Veelasha Moonsamy
 Lehrende: Dr. Veelasha Moonsamy,
 Dr. Asia Biega
 Dr. Yixin Zou

Verwendung des Moduls

M.Sc. IT-Sicherheit/Informationstechnik
 M.Sc. IT-Sicherheit/Netze und Systeme

Vorkenntnisse

Recommended but not mandatory:
 Einführung in die Usable Security and Privacy (211036); Datenschutz (260081); Basic knowledge of threat modeling; General understanding of machine learning and data science

Lernziele (learning outcomes)

By the end of the course, the student will be able to:
 Reason about privacy concerns and perform threat modelling
 Apply privacy-by-design techniques for systems implementation
 Develop privacy technologies
 Understand concepts related to data governance, including data minimization
 Design privacy-friendly, usable systems
 Understand concept related to UX design & usable privacy

Inhalt

This course will provide students with the knowledge and applied skills to tackle the design and implementation of privacy-preserving systems. Students will gain a critical understanding of privacy's role in society and tensions between privacy, technology and security. Students will learn to analyze privacy issues and develop privacy-friendly solutions by considering social, technical, legal and public policy aspects. The course includes mandatory lecture attendance, readings and group project.

The course will cover the following topics:
 Privacy definitions and concepts
 Privacy by design
 Privacy engineering: design and evaluation
 Data governance
 Notion of "Right to be forgotten"
 Usable privacy, including UX design
 Inclusive privacy

Lehrformen

The course includes mandatory lecture attendance, readings and group project.

Prüfungsformen

There will be one semester-long individual project and a written exam (60 minutes).

Voraussetzungen für die Vergabe von Credits

Final grade: 50% project + 50% exam. You need to pass the exam (i.e. achieve more than 50 points) in order to pass the course.

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/91: M.Sc. IT-Sicherheit/Informationstechnik [PO 22]

5/84: M.Sc. IT-Sicherheit/Informationstechnik [PO 20]

5/99: M.Sc. IT-Sicherheit/Netze und Systeme [PO 22]

5/96: M.Sc. IT-Sicherheit/Netze und Systeme [PO 20]

| | | | | | |
|--|------------------------|--------------------------|---|---------------------------------|------------------------------------|
| Titel des Moduls: Processor Security Processor Security | | | | | |
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung | Turnus Sommersemester | Dauer Semester |
| Lehrveranstaltungen Processor Security (211099) | | | Kontaktzeit 60 h | Selbststudium 90 h | Gruppengröße Studierende |
| Unterrichtssprache Deutsch | | | Teilnahmevoraussetzungen | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Dr.-Ing. Pascal Sasdrich Lehrende: Dr.-Ing. Pascal Sasdrich | | | | | |
| Verwendung des Moduls B.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Netze und Systeme | | | | | |
| Vorkenntnisse Inhalte der Module Informatik 1 - Programmierung und Technische Informatik 1 - Rechnerarchitektur | | | | | |
| Lernziele (learning outcomes) Im Rahmen dieser Veranstaltung lernen die Studierenden wichtige Sicherheitsaspekte und -konzepte moderner Prozessoren kennen. Der Fokus der Veranstaltung liegt dabei auf (a) Kenntnis gängiger Angriffsvektoren, (b) Verständnis der zugrundeliegenden Hardware- und Prozessormechanismen, (c) Diskussion möglicher Gegenmaßnahmen, sowohl in Hardware als auch Software. | | | | | |
| Inhalt Moderne Prozessorenarchitekturen, von eingebetteten Mikrocontrollern bis hin zu Server-CPU's, bilden das Kernstück unserer heutigen Informationsgesellschaft und werden seit Jahrzehnten immer komplizierter. Diese gesteigerte Komplexität führt aber unausweichlich zu neuen Schwachstellen und gesteigerter Anfälligkeiten gegen gezielte Angriffe. Im Rahmen dieser Veranstaltung werden daher verschiedene Sicherheitsaspekte und -konzepte moderner Prozessorarchitekturen vorgestellt und erläutert. Dazu werden sowohl wichtige Angriffsvektoren (z.B. Buffer Overflows, Privilege Escalation, Control-Flow Manipulation, Side Channel Attacks, Microarchitectural Attacks, ...), fundamentale Ursachen in der Prozessorarchitektur, als auch mögliche Abwehrstrategien diskutiert. Vorlesungsbegleitend wird ein Moodle-Kurs angeboten, der zusätzliche Inhalte sowie die praktischen Übungen bereithält. | | | | | |
| Lehrformen Vorlesung mit Übung | | | | | |
| Prüfungsformen Schriftliche Modulabschlussprüfung (120 Minuten) | | | | | |
| Voraussetzungen für die Vergabe von Credits | | | | | |
| Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) | | | | | |

5/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

5/84: M.Sc. IT-Sicherheit/ Informationstechnik [PO 20]

5/99: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 22]

5/96: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 20]

| | | | | | |
|---|------------------------|--------------------------|---|---------------------------------|---------------------------------------|
| Titel des Moduls: Programmanalyse [M.Sc.] Program Analysis | | | | | |
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung / see examination regulations | Turnus Sommersemester | Dauer 1 Semester |
| Lehrveranstaltungen Programmanalyse (211015) | | | Kontaktzeit 4 SWS (60 h) | Selbststudium 90h | Gruppengröße 40 Studierende |
| Unterrichtssprache Englisch | | | Teilnahmevoraussetzungen keine | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Kevin Borgolte Lehrende: Prof. Kevin Borgolte | | | | | |
| Verwendung des Moduls M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Netze und Systeme M.Sc. Computer Science | | | | | |
| Vorkenntnisse Erfahrung in systemnaher Programmierung, Assembler sowie Programmieren in C sind hilfreich für das Verständnis der vermittelten Themen. Vorkenntnisse aus den Vorlesungen Systemsicherheit/Betriebssystemicherheit sind hilfreich aber nicht notwendig zum Verständnis der Themen. | | | | | |
| Lernziele (learning outcomes) Die Studierenden kennen verschiedene Konzepte, Techniken und Tools aus dem Bereich der Programmanalyse. Dies beinhaltet den Überblick über verschiedene Konzepte aus dem Bereich Reverse Engineering sowie Binäranalyse. Die Studierenden haben grundlegendes Verständnis von sowohl statischen als auch dynamischen Methoden zur Analyse eines gegebenen Programms. Sie sind in der Lage, verschiedene Aspekte der Programmanalyse zu beschreiben und auf neue Problemstellungen anzuwenden. | | | | | |
| Inhalt In der Vorlesung werden unter anderem die folgenden Themen und Techniken aus dem Bereich der Programmanalyse behandelt: <ul style="list-style-type: none"> • Statische und dynamische Analyse von Programmen • Analyse von Kontroll- und Datenfluss • Symbolische Ausführung • Taint Tracking • Binary Instrumentation • Program Slicing • Überblick zu existierenden Analysetools Daneben wird im ersten Teil der Vorlesung eine Einführung in x86/x64 Assembler gegeben sowie die grundlegenden Techniken aus dem Themenbereich Reverse Engineering vorgestellt. Begleitet wird die Vorlesung von Übungen, in denen die vorgestellten Konzepte und Techniken praktisch eingeübt werden. | | | | | |
| Lehrformen Vorlesung mit Übung | | | | | |
| Prüfungsformen mündliche oder schriftliche Modulabschlussprüfung (wird zu Beginn des Semesters bekanntgegeben), Anmeldung: FlexNow | | | | | |

Voraussetzungen für die Vergabe von Credits

Bestandene Modulabschlussprüfung

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

5/84: M.Sc. IT-Sicherheit/ Informationstechnik [PO 20]

5/99: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 22]

5/96: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 20]

5/97: M.Sc. Computer Science

Titel des Moduls: Proofs are programs [M.Sc.]

Proofs are programs [M.Sc.]

| | | | | | |
|--|------------------------|--------------------------|---|---------------------------------|---------------------------------------|
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung / see examination regulations | Turnus Wintersemester | Dauer 1 Semester |
| Lehrveranstaltungen Proofs are Programs (211003) | | | Kontaktzeit 60 h | Selbststudium 90 h | Gruppengröße 40 Studierende |
| Unterrichtssprache Englisch | | | Teilnahmevoraussetzungen keine | | |

Modulbeauftragte/r und hauptamtlich Lehrende

Modulbeauftragte/r: Dr. Catalin Hritcu

Lehrende: Dr. Catalin Hritcu

Dr. Clara Schneidewind

Verwendung des Moduls

M.Sc. IT-Sicherheit/ Informationstechnik

M.Sc. IT-Sicherheit/ Netze und Systeme

M.Sc. Computer Science

Vorkenntnisse

The lecture is intended for a broad range of students, from motivated BSc students to MSc and PhD students. No specific prior knowledge in logic, programming, functional programming, or programming languages is assumed, though a degree of mathematical maturity is helpful.

Lernziele (learning outcomes)

After successful completion of this course, students will be able to

- develop purely functional programs using recursive functions on numbers, lists, maps, and various kinds of trees, including the abstract syntax trees of programs;
- use functional programming concepts such as type polymorphism and higher-order functions, which are increasingly becoming mainstream;
- formally state and prove theorems in the Coq proof assistant;
- apply different proof techniques in Coq (e.g. equational reasoning, contradiction, case analysis, induction on natural numbers, lists, and trees, induction on rule derivations, proof automation);
- define new inductive types and relations in Coq and prove statements about them;
- write simple proof terms and understand the connection between constructive logics and typed functional programming that is at the heart of Coq, in which propositions are types and proofs are programs;
- comprehend how the syntax and semantics of simple imperative programs can be formally defined in Coq and how to prove theorems about such programs and languages;
- understand how the absence of information leaks can be formalized as a security property called noninterference and enforced using secure-multi execution or simple type systems.

Inhalt

Complex proofs on paper are difficult to construct, check, and maintain. This holds not only for interesting proofs in mathematics, but also for complex formal proofs about interesting programs. For this reason, machine-checked proofs created with the help of interactive tools called proof assistants are gaining increased traction in academia and industry. Proof assistants have been used to prove the correctness and security of realistic compilers, operating systems, cryptographic libraries, or smart contracts, and also to construct machine-checked proofs for challenging theorems in mathematics.

This course introduces the Coq proof assistant [3] and explains how to use it to prove properties about functional programs and inductive relations, how to formally define a simple imperative programming language, and how to

securely enforce information-flow control for functional and imperative programs. The Coq proof assistant enables us to program formal proofs interactively and it machine-checks the correctness of the proofs along the way. The design of the Coq proof assistant itself exploits a beautiful connection between programs in typed functional programming languages and proofs in constructive logics, which is known as the Curry-Howard Correspondence [4]. This deep connection between programs and proofs should make this course interesting to not only to computer scientists, but also to mathematicians and other scientists. The goal is to demystify proofs as just programs in an elegant programming language, for which the course provides a gentle introduction. The course also shows that proofs are not only a way to convince a human reader, but they can actually be fully formalized in a proof assistant like Coq and automatically checked by a computer.

This hands-on course is based on the Logical Foundations [1] and Security Foundations [2] online textbooks, which are themselves formalized and machine-checked in the Coq proof assistant. The many exercises in each book chapter are to be solved weekly mostly in Coq, from easy exercises allowing the students to practice concepts from the lecture, building incrementally to slightly more interesting programs and proofs and also to various optional challenges. Finally, this course serves as the base for a more advanced course on “Foundations of Programming Languages, Verification, and Security”.

Lehrformen

This course consists of lectures and weekly exercises, in which the students will solve problems using the Coq proof assistant for which they can get help from a tutor.

Prüfungsformen

Written final exam (120 minutes).

Voraussetzungen für die Vergabe von Credits

Passing the final written exam.

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

5/99: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 22]

5/97: M.Sc. Computer Science

Titel des Moduls: Provable Security - Promises and Misconceptions

Provable Security - Promises and Misconceptions

| | | | | | |
|--|------------------------|--------------------------|---|---------------------------------|---------------------------------------|
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung / see examination regulations | Turnus Sommersemester | Dauer 1 Semester |
| Lehrveranstaltungen | | | Kontaktzeit 60 (4 SWS) | Selbststudium 90h | Gruppengröße 30 Studierende |
| Unterrichtssprache Deutsch | | | Teilnahmevoraussetzungen Keine | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Dr.-Ing. Jakob Feldtkeller Lehrende: Dr.-Ing. Jakob Feldtkeller | | | | | |
| Verwendung des Moduls M.Sc. IT-Sicherheit / Informationstechnik M.Sc. IT-Sicherheit / Netze und Systeme | | | | | |
| Vorkenntnisse Grundlagen der Kryptographie (Modul Kryptographie) und Wahrscheinlichkeitsrechnung. Kenntnisse über Seitenkanal- und Fehlerangriffe (z. B. über das Modul Physical Attacks and Countermeasures) sind hilfreich, aber nicht erforderlich. | | | | | |
| Lernziele (learning outcomes) Die Studenten <ul style="list-style-type: none">• kennen formale Sicherheitsmodelle und grundlegende Sicherheitsdefinitionen.• können Sicherheitsbeweise verstehen, auf deren Korrektheit hin analysieren und eigene Beweise innerhalb eines Sicherheitsmodells führen.• verstehen die Kluft zwischen Theorie und Praxis und können die Angemessenheit, die Limitierungen sowie die gegebenen Garantien von Sicherheitsmodellen für Systeme in der realen Welt evaluieren. | | | | | |
| Inhalt Obwohl es einfach ist Sicherheit intuitiv zu erfassen, ist es überraschend anspruchsvoll, Sicherheitseigenschaften präzise zu definieren und zu analysieren. Theoretische Sicherheitsmodelle begegnen dieser Herausforderung, indem sie das Systemverhalten, die Fähigkeiten von Angreifern und Bedingungen für erfolgreiche Angriffe formal definieren. Dieses Modul bietet einen tiefen Einblick in Methoden der beweisbaren Sicherheit, die mithilfe formaler Modelle und Beweistechniken mathematische Sicherheitsgarantien bietet. Die Studierenden lernen mit zentralen Sicherheitsmodellen zu arbeiten, entwickeln und interpretieren Beweise und gewinnen so ein praxisnahes Verständnis der relevanten Konzepte. Hierbei liegt ein Schwerpunkt auf der kritischen Untersuchung von Grenzen der beweisbaren Sicherheit, insbesondere der Kluft zwischen theoretischen Sicherheitsbeweisen und praktischen Implementierungen in der realen Welt. Dafür werden klassische Sicherheitsmodelle um physikalische Aspekte erweitert, um physische Bedrohungen wie Seitenkanal- und Fehlerangriffe zu berücksichtigen, die traditionelle Annahmen in Sicherheitsbeweisen unterlaufen. So erlangen die Studierenden ein differenziertes Bild der Stärken und Schwächen beweisbarer Sicherheit und ihrer Bedeutung für das Design sicherer Systeme in der Praxis. | | | | | |
| Lehrformen Vorlesung mit Übung und Hausaufgaben | | | | | |
| Prüfungsformen Schriftliche Modulabschlussprüfung über 120 Minuten | | | | | |

Voraussetzungen für die Vergabe von Credits

Bestandene Modulabschlussprüfung

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/91: M.Sc. IT-Sicherheit/Informationstechnik

5/99: M.Sc. IT-Sicherheit/Netze und Systeme

Titel des Moduls: Public Key Kryptanalyse 1 [M.Sc] (nicht im SoSe 25)
Public Key Cryptanalysis 1

| | | | | | |
|---|------------------------|--------------------------|---|---------------------------------|---------------------------------------|
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung / see examination regulations | Turnus Sommersemester | Dauer 1 Semester |
| Lehrveranstaltungen Public Key Kryptanalyse 1 (211055) | | | Kontaktzeit 3 SWS (45 h) | Selbststudium 105 h | Gruppengröße 20 Studierende |
| Unterrichtssprache Deutsch | | | Teilnahmevoraussetzungen keine | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Alex May Lehrende: Prof. Alex May | | | | | |
| Verwendung des Moduls M.Sc. IT-Sicherheit / Informationstechnik M.Sc. IT-Sicherheit / Netze und Systeme M.Sc. Computer Science | | | | | |
| Vorkenntnisse Vorausgesetzt werden elementare Kenntnisse der Lineare Algebra (Mathematik 1) und Informatiker) und ein Interesse an algorithmischen Techniken und Kryptographie, in Theorie und Praxis (umgesetzt mit Hilfe des Computeralgebra-Systems Sage). | | | | | |
| Lernziele (learning outcomes) Die Studierenden sollen breite Kenntnisse zu algorithmischen Techniken der asymmetrischen Kryptanalyse, insbesondere für codierungsbasierte Kryptographie, erlangen. Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • kennen die Studierenden grundlegende Schlüsselfindungs-Algorithmen wie Brute-Force und Meet-in-the-Middle und können diese auf neue kryptographische Systeme anwenden, • beherrschen sie die Grundlagen linearer Codes und ihrer Dualcodes, insbesondere als kryptographische Anwendung das McEliece-Kryptosystem, • kennen Studierende Time-Memory Techniken wie Pollard Rho und Parallel Collision Search, und können sie auf neue Probleme anwenden, • haben Studierende einen Überblick über alle aktuellen Dekodieralgorithmen im Bereich des Information Set Decoding, die für die Sicherheits-Evaluierung moderner kodierungsbasierter Kryptosysteme relevant sind, • erlernen Studierende weiterführende Techniken für Speedups mit Hilfe von Quantenrechnern, • sind Studierende in der Lage, Techniken der Kryptanalyse mit Hilfe der Computer-Algebra Sage zu implementieren. | | | | | |
| Inhalt Kryptanalyse dient dazu, kryptographische Systeme derart zu instantiiieren, dass sie einerseits ein vordefiniertes Sicherheitsniveau bieten, andererseits aber möglichst performant sind. Die Kryptanalyse bietet dazu einen ganzen Werkzeugkoffer an algorithmischen Techniken, um die Evaluation neuer kryptographischer Systeme zu realisieren. Dies beinhaltet sowohl klassische Algorithmen als auch Algorithmen für Quantenrechner, damit die verwendete Kryptographie selbst in einer Ära von Quantenrechnern sicher bleiben. | | | | | |

Lehrformen

Die Vorlesung wird als seminaristischer Unterricht abgehalten, die praktischen Übungen am Rechner mit der Computer-Algebra Sage werden zudem weitere Lehrformen wie Gruppen- und Projektarbeit beinhalten.

Prüfungsformen

Schriftliche Modulabschlussprüfung über 120 Minuten

Voraussetzungen für die Vergabe von Credits

Bestandene schriftliche Modulabschlussprüfung

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/91 M.Sc IT-Sicherheit/ Informationstechnik [PO22]

5/84 M.Sc. IT-Sicherheit/ Informationstechnik [PO20]

5/99 M.Sc IT-Sicherheit/ Netze und Systeme [PO22]

5/96 M.Sc IT-Sicherheit/ Netze und Systeme [PO20]

5/97: M.Sc. Computer Science

| | | | | | |
|--|------------------------|--------------------------|---|---------------------------------|------------------------------------|
| Titel des Moduls: Public Key Verschlüsselung | | | | | |
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung | Turnus Wintersemester | Dauer 1 Semester |
| Lehrveranstaltungen Public Key Verschlüsselung | | | Kontaktzeit 60 h | Selbststudium 90 h | Gruppengröße Studierende |
| Unterrichtssprache Englisch | | | Teilnahmevoraussetzungen | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Jun. Prof. Dr. Nils Fleischhacker Lehrende: Jun. Prof. Dr. Nils Fleischhacker | | | | | |
| Verwendung des Moduls M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Netze und Systeme | | | | | |
| Vorkenntnisse Als Voraussetzung für die Vorlesung sind Vorkenntnisse in Kryptographie und beweisbarer Sicherheit, insbesondere von Reduktionsbeweisen, hilfreich aber nicht zwingend erforderlich. | | | | | |
| Lernziele (learning outcomes) Die Studierenden haben einen Einblick in in theoretische und praktische Aspekte der Public Key Verschlüsselung erhalten | | | | | |
| Inhalt Die Vorlesung gibt einen Einblick in theoretische und praktische Aspekte der Public Key Verschlüsselung. Dies umfasst Grundlagen und formalen Definitionen von Sicherheit (CPA, CCA1, CCA2), die beweisbare Sicherheit verschiedener theoretischer und praktischer Konstruktionen, sowie die Verbindungen von Public Key Verschlüsselung zu anderen Aspekten der Kryptographie. | | | | | |
| Lehrformen Vorlesung mit Übung | | | | | |
| Prüfungsformen Mündlich (30 Minuten) | | | | | |
| Voraussetzungen für die Vergabe von Credits Bestandene Modulabschlussprüfung | | | | | |
| Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 5/91: M.Sc. IT-Sicherheit/ Informationstechnik 5/99: M.Sc. IT-Sicherheit/ Netze und Systeme | | | | | |

Titel des Moduls: Quantum Cryptography (kein Angebot im WS 25/26)
Quantum Cryptography (no offer in WS 25/26)

| | | | | | |
|---|------------------------|--------------------------|---|---------------------------------|------------------------------------|
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung / see examination regulations | Turnus Wintersemester | Dauer 1 Semester |
| Lehrveranstaltungen Quantum Cryptography (212016) | | | Kontaktzeit 4 SWS (60 h) | Selbststudium 90 h | Gruppengröße Studierende |
| Unterrichtssprache Englisch | | | Teilnahmevoraussetzungen keine | | |

Modulbeauftragte/r und hauptamtlich Lehrende

Modulbeauftragte/r: Prof. Michael Walter
 Lehrende: Prof. Michael Walter
 Dr. Giulio Malavolta

Verwendung des Moduls

M.Sc. IT-Sicherheit/ Informationstechnik
 M.Sc. IT-Sicherheit/ Netze und Systeme
 M.Sc. Computer Science

Vorkenntnisse

keine

Lernziele (learning outcomes)

You will learn fundamental concepts, algorithms, protocols, and results in quantum (and quantum-resistant) cryptography. After successful completion of this course, you will know how to generalize cryptographic concepts to the quantum setting, how quantum algorithms can attack well-known cryptographic protocols, and how to design and analyze classical and quantum protocols for protecting classical and quantum data against quantum adversaries. You will be prepared for a research or thesis project in this area.

Inhalt

This course will give an introduction to the interplay of quantum information and cryptography, which has recently led to much excitement and insights – including by researchers at CASA right here on our very own campus. We will begin with a brief introduction to both fields and discuss in the first half of the course how quantum computers can attack classical cryptography and how to overcome this challenge – either by protecting against the power of quantum computers or by leveraging the power of quantum information. In the second half of the course, we will discuss how to generalize cryptography to protect quantum data and computation.

Topics to be covered will likely include:

- * Basic quantum computing
- * Basic cryptography
- * Quantum attacks on classical cryptography
- * Quantum random oracles and compressed oracle technique
- * Quantum-resistant cryptography in light of the NIST competition

- * Classical vs quantum information
- * Quantum money
- * Quantum key distribution
- * Quantum complexity theory
- * Quantum pseudorandomness
- * From classical to quantum fully homomorphic encryption
- * Classical verification of quantum computation
- * Quantum rewinding

This course should be of interest to students of computer science, mathematics, physics, and related disciplines. Students interested in a Master's project in quantum or quantum-resistant cryptography, quantum information, quantum computing, and similar are particularly encouraged to participate.

Lehrformen

Vorlesung mit Übungen

Prüfungsformen

Modulabschlussprüfung; schriftlich oder mündlich je nach Teilnehmendenzahl.

Voraussetzungen für die Vergabe von Credits

Bestandene Modulabschlussprüfung

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/91 M.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

5/84 M.Sc. IT-Sicherheit/ Informationstechnik [PO 20]

5/99 :M.Sc. IT-Sicherheit/ Netze und Systeme [PO 22]

5/96 :M.Sc. IT-Sicherheit/ Netze und Systeme [PO 20]

5/97: M.Sc. Computer Science

Titel des Moduls: Quantum Information and Computation [M.Sc.]
Quantum Information and Computation [M.Sc.]

| | | | | | |
|--|------------------------|--------------------------|---|---------------------------------|------------------------------------|
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung / see examination regulations | Turnus Wintersemester | Dauer 1 Semester |
| Lehrveranstaltungen Quantum Information and Computation (212011) | | | Kontaktzeit 4 SWS (60 h) | Selbststudium 90 h | Gruppengröße Studierende |
| Unterrichtssprache Deutsch oder Englisch (depends on audience) | | | Teilnahmevoraussetzungen Keine | | |

Modulbeauftragte/r und hauptamtlich Lehrende

Modulbeauftragte/r: Prof. Dr. Michael Walter
 Lehrende: Prof. Dr. Michael Walter

Verwendung des Moduls

M.Sc. Angewandte Informatik
 M.Sc. IT-Sicherheit/ Informationstechnik
 M.Sc. IT-Sicherheit/ Netze und Systeme
 M.Sc. Computer Science

Vorkenntnisse

Familiarity with linear algebra (in finite dimensions) and probability (with finitely many outcomes) at the level of a first Bachelors course; we will briefly remind you of the more difficult bits in class. In addition, some mathematical maturity, since we will discuss precise mathematical statements and rigorous proofs. No background in physics is required.

Lernziele (learning outcomes)

You will learn fundamental concepts, algorithms, and results in quantum information and computation. After successful completion of this course, you will know the theoretical model of quantum information and computation, how to generalize computer science concepts to the quantum setting, how to design and analyze quantum algorithms and protocols for a variety of computational problems, and how to prove complexity theoretic lower bounds. You will be prepared for an advanced course or a research or thesis project in this area. Master's students will be expected to understand the material in a deeper way, which will reflect itself in homework and examination.

Inhalt

This course will give an introduction to quantum information and quantum computation from the perspective of theoretical computer science.

Topics to be covered will likely include:

- Fundamentals of quantum computing: quantum bits, states and operations
- The power of quantum entanglement: nonlocal games
- Entanglement as a resource: superdense coding and teleportation
- Quantum circuit model of computation
- Quantum computing with oracles: Deutsch-Jozsa, Bernstein-Vazirani, Simon
- Quantum Fourier transform and phase estimation
- Shor's factoring algorithm
- Grover's search algorithm and beyond: how to solve SAT on a quantum computer?
- From no cloning to quantum money: a peek at quantum cryptography

The course should be of interest to students of computer science, mathematics, physics, and related disciplines. Students interested in a BSc or MSc project in quantum information, computing, cryptography, etc. are particularly

encouraged to participate.

Lehrformen

Lecture with Exercise

Prüfungsformen

Final written module exam (180 minutes)

Voraussetzungen für die Vergabe von Credits

Passed written exam

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/105: M.Sc. Angewandte Informatik

5 /91: M.Sc. IT-Sicherheit/ Informationstechnik

5/ 99: M.Sc. IT-Sicherheit/ Netze und Systeme

5/97: M.Sc. Computer Science

| | | | | | |
|--|------------------------|--------------------------|---|---------------------------------|------------------------------------|
| Titel des Moduls: Software Protection Software Protection | | | | | |
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung | Turnus Sommersemester | Dauer 1 Semester |
| Lehrveranstaltungen Software Protection (211107) | | | Kontaktzeit 45 h | Selbststudium 105 h | Gruppengröße Studierende |
| Unterrichtssprache Deutsch | | | Teilnahmevoraussetzungen keine | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Studiendekan IT-Sicherheit Lehrende: Dr.-Ing. Tim Blazytko Philipp Koppe | | | | | |
| Verwendung des Moduls B.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Netze und Systeme | | | | | |
| Vorkenntnisse Im Bereich Reverse Engineering sind empfohlen, beispielsweise durch Erfahrung mit x86-Assembler. Erfahrung in systemnaher Programmierung (Assembler, C) ist hilfreich. | | | | | |
| Lernziele (learning outcomes) Die Studierenden kennen verschiedene Konzepte, Techniken und Tools aus dem Bereich Software Protection. Dies beinhaltet sowohl Wissen über das Design und die Implementierung von Obfuskerungstechniken als auch die Sicherheitsanalyse gängiger Systeme. Die Studierenden lernen erweiterte Techniken zur Programmanalyse, mit welchen sie komplexe Protection-Mechanismen angreifen können. Sie sind in der Lage, verschiedene Aspekte der Software Protection zu beschreiben und auf neue Problemstellungen anzuwenden. | | | | | |
| Inhalt Unter Software Protection versteht man Maßnahmen, welche die Analyse bzw. das Reverse Engineering von Software erschweren. Solche Methoden finden sowohl Anwendung in kommerzieller Software, um Piraterie zu verhindern, als auch in Malware, um deren Funktionsweise zu verschleiern. In dieser Lehrveranstaltung lernen die Studierenden gängige Methoden der Software Protection kennen sowie Methoden, um diese zu brechen. Dazu designen und implementieren sie in praxisnahen Aufgaben erst ihre eigenen Protection-Mechanismen, welche sie im Anschluss brechen werden mit dem Ziel, diese wieder zu verbessern. Parallel dazu werden Schutzmechanismen aus der echten Welt analysiert, attackiert und diskutiert. Dabei werden unter anderem die folgenden Themen und Techniken aus dem Bereich Software Protection behandelt: - Opaque Predicates - Control-flow Flattening - Mixed Boolean-Arithmetic Expressions - Virtual Machines | | | | | |

- Anti-Tamper
- Symbolische Ausführung
- SMT Solving
- Programmsynthese
- Überblick zu existierenden Analysetools und Frameworks

Lehrformen

Vorlesung mit Übung

Prüfungsformen

Arbeit/Kompetenznachweis im Semester. Die Lehrveranstaltung beinhaltet mehrere benotete praktische Übungen mit einer Dauer von 2-3 Wochen pro Übung. Jeder Teilnehmer bearbeitet die Übungen selbstständig in Einzelarbeit. Die Modulabschlussnote bildet sich aus dem gewichteten arithmetischen Mittel der einzelnen Übungen.

Voraussetzungen für die Vergabe von Credits

Erfolgreiche Kompetenznachweis im Semester

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/150: Bachelor IT-Sicherheit/ Informationstechnik [PO 22]

5/149: Bachelor IT-Sicherheit/ Informationstechnik [PO 20]

5/91: Master IT-Sicherheit/ Informationstechnik [PO 22]

5/84: Master IT-Sicherheit/ Informationstechnik [PO 20]

5/99: Master IT-Sicherheit/ Netze und Systeme [PO 22]

5/96: Master IT-Sicherheit/ Netze und Systeme [PO 20]

Titel des Moduls: Software Security 1 [M.Sc.]
Software Security 1 [M.Sc.]

| | | | | | |
|--|------------------------|--------------------------|---|---------------------------------|------------------------------------|
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung / see examination regulations | Turnus Wintersemester | Dauer 1 Semester |
| Lehrveranstaltungen Software Security 1 (212026) | | | Kontaktzeit 4 SWS (60 h) | Selbststudium 90 h | Gruppengröße Studierende |
| Unterrichtssprache Englisch | | | Teilnahmevoraussetzungen | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Kevin Borgolte Lehrende: Prof. Kevin Borgolte | | | | | |
| Verwendung des Moduls M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Netze und Systeme M.Sc. Angewandte Informatik M.Sc. Computer Science | | | | | |
| Vorkenntnisse Prior knowledge about programming in Python, C, and assembler is recommended. The following courses (or equivalent) are required: System Security (211011) Operating Systems (211005) | | | | | |
| Lernziele (learning outcomes) At the end of this course, students will be able to: <ul style="list-style-type: none"> • analyze user-space software vulnerability types and protection mechanisms • understand how to write code to reduce the risk of vulnerabilities and apply defensive programming techniques • identify new software vulnerabilities and evaluate their impact • show the existence of a vulnerability, for example, by developing proof of concept verifications | | | | | |
| Inhalt The course covers the area of introductory software security, vulnerability discovery, and vulnerability verification, focusing on: <ul style="list-style-type: none"> • Assembly and Disassembly, Shellcode • Binary Reverse Engineering and Debugging • Memory and Type Safety/Errors • Stack-based Buffer Overflows • Heap Attacks • Information Leakage • Format String Vulnerabilities • Code Re-use Attacks • Types and Type Safety • Race Conditions | | | | | |
| Lehrformen Vorlesung mit Übung | | | | | |

Prüfungsformen

Practical exam.

Voraussetzungen für die Vergabe von Credits

Passed practical exam. The exam will take place on two days of 6 hours each, both dates are necessary to pass.

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/91: M.Sc. IT-Sicherheit/ Informationstechnik

5/99: M.Sc. IT-Sicherheit/ Netze und Systeme

5/105: M.Sc. Angewandte Informatik

5/91: M.Sc. Computer Science

Titel des Moduls: Software Security 2
Software Security 2

| | | | | | |
|-----------------------|------------------------|--------------------------|---|---------------------------------|----------------------------|
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung / see examination regulations | Turnus Sommersemester | Dauer 1 Semester |
|-----------------------|------------------------|--------------------------|---|---------------------------------|----------------------------|

| | | | |
|--|-----------------------------------|------------------------------|------------------------------------|
| Lehrveranstaltungen Software Security 2 (211063) | Kontaktzeit 60h (4 SWS) | Selbststudium 90 h | Gruppengröße Studierende |
|--|-----------------------------------|------------------------------|------------------------------------|

| | |
|---------------------------------------|---------------------------------|
| Unterrichtssprache Englisch | Teilnahmevoraussetzungen |
|---------------------------------------|---------------------------------|

Modulbeauftragte/r und hauptamtlich Lehrende
 Modulbeauftragte/r: Prof. Kevin Borgolte
 Lehrende: Prof. Kevin Borgolte

Verwendung des Moduls
 M.Sc. IT-Sicherheit/Informationstechnik
 M.Sc. IT-Sicherheit / Netze und Systeme
 M.Sc. Computer Science

Vorkenntnisse
 Prior knowledge about programming in Python, C, and assembler is highly recommended. The following courses (or equivalent) are required:
 System Security (211011)
 Operating Systems (211005)
 Software Security 1 (212026)

Lernziele (learning outcomes)
 At the end of this course, students will be able to:

- classify and describe complex vulnerabilities and advanced protection mechanisms of a diverse set of software systems
- analyze and reason about protection mechanisms for modern software systems across its layers from userspace to kernel to hypervisor
- identify end-to-end vulnerabilities in software systems
- develop proofs of concept exploits/verifications to show the existence of an end-to-end vulnerability in a modern software system with modern defenses
- understand how to write code defensively to reduce the risk of vulnerabilities

Inhalt
 The course covers the area of advanced topics in software security, vulnerability discovery, and vulnerability verification, focusing on:

- Attacks on Just-in-time Compilers
- Sandboxing Techniques
- Browser Vulnerabilities
- Kernel and Hypervisor Vulnerabilities
- Non-x86 Architectures
- Non-Linux Operating Systems
- Automated Exploit/Verification Synthesis

Lehrformen
 Vorlesung mit Übung

Prüfungsformen

Praktische Prüfung.

Voraussetzungen für die Vergabe von Credits

Passed practical exam. The exact schedule and details will be communicated during the first lecture.

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/91: M.Sc. IT-Sicherheit/Informationstechnik

5/99: M.Sc. IT-Sicherheit / Netze und Systeme

5/97: M.Sc. Computer Science

| | | | | | |
|--|------------------------|--------------------------|---|---------------------------------|------------------------------------|
| Titel des Moduls: Software-Implementierung kryptographischer Verfahren | | | | | |
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung | Turnus Sommersemester | Dauer 1 Semester |
| Lehrveranstaltungen Software-Implementierung kryptographischer Verfahren (211035) | | | Kontaktzeit 60 h | Selbststudium 90 h | Gruppengröße Studierende |
| Unterrichtssprache Deutsch | | | Teilnahmevoraussetzungen keine | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr.-Ing. Tim Guneyusu Lehrende: Dr.-Ing. Max Hoffmann | | | | | |
| Verwendung des Moduls M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Netze und Systeme | | | | | |
| Vorkenntnisse Grundkenntnisse der Programmiersprache C bzw. C++, Vorlesung „Einführung in die Kryptographie I“ | | | | | |
| Lernziele (learning outcomes) Die Studierenden haben ein Verständnis für Methoden für die schnelle Software-Realisierung ausgewählter Krypto-Verfahren und diese selbst implementiert. | | | | | |
| Inhalt Es werden ausgewählte fortgeschrittene Implementierungstechniken der modernen Kryptographie behandelt. Inhalte: <ul style="list-style-type: none"> • Effiziente Implementierung von Blockchiffren • Bitslicing • Effiziente Arithmetik in $GF(2^m)$ • Effiziente Arithmetik auf elliptischen Kurven • Spezielle Primzahlen zur schnellen modularen Reduktion • Primzahltests • Post-Quantum Kryptographie • Secure Coding | | | | | |
| Lehrformen Vorlesung mit Übungen | | | | | |
| Prüfungsformen Schriftliche Modulabschlussprüfung (120 Minuten) und Projektarbeit. Die Klausur geht mit 80% und das Projekt mit 20% in die Modulnote ein. | | | | | |
| Voraussetzungen für die Vergabe von Credits Es müssen mindestens 50 Prozent aller möglichen Punkte in der Klausur und den semesterbegleitenden Projekten erreicht werden. | | | | | |
| Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 5/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22] | | | | | |

5/84: M.Sc. IT-Sicherheit/ Informationstechnik [PO 20]

5/99: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 22]

5/96: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 20]

| | | | | | |
|---|------------------------|--------------------------|---|---------------------------------|------------------------------------|
| Titel des Moduls: Symmetrische Kryptanalyse | | | | | |
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung | Turnus Wintersemester | Dauer 1 Semester |
| Lehrveranstaltungen Symmetrische Kryptanalyse | | | Kontaktzeit 60 h | Selbststudium 90 h | Gruppengröße Studierende |
| Unterrichtssprache Deutsch | | | Teilnahmevoraussetzungen | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Nils-Gregor Leander Lehrende: Prof. Dr. Nils-Gregor Leander | | | | | |
| Verwendung des Moduls M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Netze und Systeme | | | | | |
| Vorkenntnisse <p>Inhalt der Vorlesung "Einführung in die Kryptographie 1" | | | | | |
| Lernziele (learning outcomes) Die Studierenden haben ein vertieftes Verständnis für die Sicherheit symmetrischer Chiffren. | | | | | |
| Inhalt Wir behandeln die wichtigsten Themen in der symmetrischen Kryptanalyse. Nach einer ausführlichen Vorstellung von linearer und differentieller Kryptanalyse werden weitere Angriffe auf symmetrische Primitive, insbesondere Block-Chiffren behandelt. Hierzu zählen insbesondere Integral (auch Square) Attacks, Impossible Differentials, Boomerang-Angriffe und Slide-Attacks. Neben den Angriffen selbst werden auch immer die daraus resultierenden Design-Kriterien beschrieben, um neue Algorithmen sicher gegen die Angriffe zu machen. | | | | | |
| Lehrformen | | | | | |
| Prüfungsformen Mündliche Modulabschlussprüfung (30 Minuten) | | | | | |
| Voraussetzungen für die Vergabe von Credits Bestandene mündliche Modulabschlussprüfung. | | | | | |
| Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 5/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22] 5/84: M.Sc. IT-Sicherheit/ Informationstechnik [PO 20] 5/99: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 22] 5/96: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 20] | | | | | |

| | | | | | |
|--|------------------------|-------------------------|---------------------------------|---------------------------------|------------------------------------|
| Titel des Moduls: Vertiefungsseminar (M.Sc. IT-Sicherheit) | | | | | |
| Modul-Nr./Code | Credits 3 CP | Workload 90 h | Semester | Turnus jedes Semester | Dauer Semester |
| Lehrveranstaltungen In jedem Semester wird eine wechselnde Auswahl an Seminaren bereitgestellt. Die zugeordneten Seminare können im Vorlesungsverzeichnis eingesehen werden. | | | Kontaktzeit 30 h | Selbststudium | Gruppengröße Studierende |
| Unterrichtssprache Deutsch oder Englisch | | | Teilnahmevoraussetzungen | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Studiendekan IT-Sicherheit Lehrende: siehe jeweiliges Seminar | | | | | |
| Verwendung des Moduls M.Sc. IT-Sicherheit / Informationstechnik | | | | | |
| Vorkenntnisse Die Vertiefungsseminare beziehen sich in der Regel auf Inhalte aus bestimmten Pflicht- oder Vertiefungsmodulen, die im Vorfeld absolviert worden sein sollten. | | | | | |
| Lernziele (learning outcomes) Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • verfügen Studierende über vertiefte wissenschaftliche Kenntnisse in dem ausgewählten Seminarthema • haben Studierende das Halten eines wissenschaftlichen Vortrags praktisch eingeübt und können Forschungsergebnisse eigenständig in einem didaktisch wohl aufbereiteten Vortrag vermitteln • können die Teilnehmer konstruktives Feedback formulieren und entgegennehmen • können Studierende eine schriftliche Ausarbeitung zu ihrem Seminarvortrag verfassen | | | | | |
| Inhalt Es werden Masterseminare zu mehreren relevanten Themen aus der IT-Sicherheit angeboten, wie beispielsweise zu Netz- und Datensicherheit, Implementation Security, Human Centred Security and Privacy oder Kryptographie. Von den angebotenen Themen wählen die Studierenden abhängig von den eigenen Interessen und den individuellen Vertiefungswünschen ein Thema aus. Dieses sollen die Studierenden selbstständig bearbeiten. Dazu gehören die Literaturrecherche, die Einarbeitung in das Thema und schließlich die Präsentation. Nähere Informationen sind zu den jeweiligen Seminaren im Vorlesungsverzeichnis zu entnehmen. | | | | | |
| Lehrformen Seminar | | | | | |
| Prüfungsformen Seminarvortrag | | | | | |
| Voraussetzungen für die Vergabe von Credits | | | | | |
| Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 0/Summe der prüfungsrelevanten CP [PO 20] 3/Summe der prüfungsrelevanten CP [PO 22] | | | | | |

| | | | | | |
|--|------------------------|--------------------------|---|---------------------------------|------------------------------------|
| Titel des Moduls: Zero-Knowledge Proof Systems Zero-Knowledge Proof Systems | | | | | |
| Modul-Nr./Code | Credits 5 CP | Workload 150 h | Semester siehe Prüfungsordnung | Turnus Sommersemester | Dauer 1 Semester |
| Lehrveranstaltungen Ze-ro-Know-ledge Proof Sys-tems (211032) | | | Kontaktzeit 60 h | Selbststudium 90 h | Gruppengröße Studierende |
| Unterrichtssprache Deutsch | | | Teilnahmevoraussetzungen keine | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Jun. Prof. Dr. Nils Fleischhacker Lehrende: Jun. Prof. Dr. Nils Fleischhacker | | | | | |
| Verwendung des Moduls M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Netze und Systeme | | | | | |
| Vorkenntnisse Einführung in die Kryptographie | | | | | |
| Lernziele (learning outcomes) A deep understanding of the Foundations and Applications of Zero-Knowledge Proof Systems. This includes an understanding of the necessary underlying assumptions, the lower bound on what is possible to achieve, as well as efficient instantiations from concrete assumptions. | | | | | |
| Inhalt Zero-Knowledge protocols are important building blocks for more complex cryptographic protocols. This class covers foundational aspects of zero-knowledge proofs, including: Lower bounds and round complexity, necessary assumptions, communication complexity, and zero-knowledge in a quantum world, as well as theoretical and practical constructions and their security proofs. Topics: Cryptography, Interactive Proof Systems, Zero-Knowledge Proofs, Provable Security | | | | | |
| Lehrformen Lecture with exercise | | | | | |
| Prüfungsformen Oral Exam | | | | | |
| Voraussetzungen für die Vergabe von Credits Bestandene Modulabschlussprüfung | | | | | |
| Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 5/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22] 5/99: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 22] | | | | | |

| | | | | | |
|--|-------------------------|-----------------|--|----------------------|------------------------------------|
| Titel des Moduls: Freie Wahlmodule free electives | | | | | |
| Modul-Nr./Code | Credits 25 CP | Workload | Semester | Turnus | Dauer Semester |
| Lehrveranstaltungen | | | Kontaktzeit siehe Lehrveranstaltungen | Selbststudium | Gruppengröße Studierende |
| Unterrichtssprache | | | Teilnahmevoraussetzungen siehe Lehrveranstaltungen | | |
| Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Lehrende: | | | | | |
| Verwendung des Moduls | | | | | |
| Vorkenntnisse | | | | | |
| Lernziele (learning outcomes) Die Studierenden beherrschen entsprechend ihrer Wahl verschiedene, das Studium ergänzende Schlüsselqualifikationen und haben ihr Fachwissen vertieft. | | | | | |
| Inhalt Durch die freie Wahl von Lehrveranstaltungen aus dem gesamten Angebot der RUB, UARuhr und UNIC können die Studierenden fachliche und überfachliche Schwerpunkte anhand ihrer eigenen Interessen setzen. Je nach Veranstaltungswahl werden unterschiedliche Inhalte vermittelt. | | | | | |
| Lehrformen | | | | | |
| Prüfungsformen | | | | | |
| Voraussetzungen für die Vergabe von Credits | | | | | |
| Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) unbenotet | | | | | |

| Titel des Moduls: Masterarbeit und Kolloquium (ITS) | | | | | |
|--|----------------|-----------------|--|----------------------|---------------------|
| Modul-Nr./Code | Credits | Workload | Semester | Turnus | Dauer |
| | 30 CP | 900 h | 4 | jedes Semester | 1 Semester |
| Lehrveranstaltungen | | | Kontaktzeit | Selbststudium | Gruppengröße |
| | | | 15h | 885 h | Studierende |
| Unterrichtssprache | | | Teilnahmevoraussetzungen | | |
| Englisch oder Deutsch | | | Erfolgreich abgeschlossene Module im Umfang von 70 CP (PO22) bzw. 80 CP (PO20) | | |
| Modulbeauftragte/r und hauptamtlich Lehrende | | | | | |
| Modulbeauftragte/r: Studiendekan IT-Sicherheit Lehrende: Lehrende im Studiengang IT-Sicherheit | | | | | |
| Verwendung des Moduls | | | | | |
| M.Sc. IT-Sicherheit / Informationstechnik M.Sc. IT-Sicherheit / Netze und Systeme | | | | | |
| Vorkenntnisse | | | | | |
| Abhängig von der Themenwahl | | | | | |
| Lernziele (learning outcomes) | | | | | |
| Nach erfolgreichem Abschluss des Moduls: | | | | | |
| <ul style="list-style-type: none"> • können Studierende selbstständig und fristgerecht ein wissenschaftliches Thema bearbeiten von der Recherche bis zur Dokumentation der Resultate • können Studierende geeignete wissenschaftliche Verfahren und Methoden, die sie im Studium kennengelernt haben, auswählen, anwenden und weiterentwickeln, um ein konkretes Problem zu lösen • können Studierende ihre Ergebnisse kritisch mit dem Stand der Forschung vergleichen und evaluieren • können Studierende ihre eigenen Ergebnisse angemessen in Wort und Schrift darstellen. | | | | | |
| Inhalt | | | | | |
| Die Masterarbeit stellt eine forschungsorientierte, sechsmonatige Arbeit zu einem bestimmten Thema aus dem Bereich der IT-Sicherheit dar und wird im letzten Semester des Studiums geschrieben. Diese hat ein Umfang von 30 Leistungspunkten. Die Masterarbeit wird auf Englisch oder Deutsch verfasst. | | | | | |
| Lehrformen | | | | | |
| Abschlussarbeit | | | | | |
| Prüfungsformen | | | | | |
| Masterarbeit und Kolloquiumsvortrag | | | | | |
| Voraussetzungen für die Vergabe von Credits | | | | | |
| Sowohl die Masterarbeit als auch der Kolloquiumsvortrag müssen bestanden sein. Der Anteil der Kolloquiumsnote an der Gesamtnote beträgt 10% | | | | | |
| Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) | | | | | |
| 30/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22] | | | | | |
| 30/84: M.Sc. IT-Sicherheit/ Informationstechnik [PO 20] | | | | | |
| 30/99: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 22] | | | | | |
| 30/96: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 20] | | | | | |