RUB Faculty of Computer Science
www.informatik.rub.de

RUHR UNIVERSITÄT BOCHUM RUB

# Modulhandbuch
# Master of Science (M.Sc.)

## IT-Sicherheit / Informatikstechnik
## [PO22]

**Stand: Wintersemester 2023/24**
**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 ≥ 15, b ≥ 15, c ≥ 15, a+b+c ≥ 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.#

| | Lehrveranstaltung | Einheit | Umfang Modul (LP) | Semester | Bewertung |
|---|---|---|---|---|---|
| Wahlpflichtmodule | | | | | |
| Theorie der IT-Sicherheit | | | | | |
| | Authentische Schlüsselvereinbarung: Formale Modelle und Anwendungen | Informatik | 5 | SS | benotet |
| | Deep Learning | Informatik | 5 | Letztmalig WS 22/23 | benotet |
| | Foundations of Programming Languages, Verification, and Security | Informatik | 5 | WS | benotet |
| | Fundamentals of Data Science | Informatik | 5 | SS | benotet |
| | Kryptographische Protokolle | Informatik | 5 | SS | benotet |
| | Proofs are programs | Informatik | 5 | SS | benotet |
| | Public Key Kryptanalyse 1 | Informatik | 5 | SS | benotet |
| | Public Key Verschlüsselung | Informatik | 5 | WS (kein Angebot im WS 23/24) | benotet |
| | Quantum Cryptography | Informatik | 5 | WS (kein Angebot im WS 23/24) | benotet |
| | Symmetrische Kryptanalyse | Informatik | 5 | WS | benotet |
| | Zero-Knowledge Proof Systems | Informatik | 5 | SS | benotet |
| Anwendungen der IT-Sicherheit | | | | | |
| | Aktuelle Themen im Bereich der Internet-Sicherheit | Informatik | 5 | WS | benotet |
| | Aufbau eines Managementsystems für Informationssicherheit nach DIN ISO / 27001 | Informatik | 4 | SS | benotet |
| | Blockchain Security and Privacy | Informatik | 5 | WS | benotet |
| | Developer Centered Security | Informatik | 5 | SS | benotet |
| | Empirische IT-Sicherheitsforschung | Informatik | 5 | WS | benotet |
| | Human Aspects of Cryptography Adoption | Informatik | 5 | WS | benotet |
| | Menschliches Verhalten in der IT-Sicherheit | Informatik | 5 | SS | benotet |
| | Message Level Security | Informatik | 5 | WS | benotet |
| | Microarchitectural Attacks and Defenses | Informatik | 5 | WS | benotet |
| | Privacy, data governace and usability | Informatik | 5 | WS | benotet |
| | Processor Security | Informatik | 5 | SS | benotet |
| | Programmanalyse | Informatik | 5 | SS | benotet |
| | Software Protection | Informatik | 5 | SS | benotet |
| | Software Security | Informatik | 9 | WS | benotet |
| | Software-Implementierung kryptographischer Verfahren | Informatik | 5 | SS | benotet |
| Informatik | | | | | |
| | Advanced Algorithms | Informatik | 9 | WS | benotet |
| | Autonomous Vehicles and Artificial Intelligence | Informatik | 5 | SS | benotet |
| | Autonomous Vehicles and Artificial Intelligence Lab | Informatik | 5 | WS | benotet |
| | Datenbanksysteme | Mathematik | 9 | Letzmalig WS 22/23 | benotet |
| | Deep Learning | Informatik | 5 | WS | benotet |
| | Deterministic Network Calculus | Informatik | 5 | SS | benotet |
| | Distributed System | Informatik | 5 | Letztmalig SS 23 | benotet |
| | Effiziente Algorithmen | Informatik | 9 | SS | benotet |
| | Embedded Multimedia | Informatik | 6 | SS | benotet |
| | Energy-Aware Computing Systems | Informatik | 6 | WS | benotet |
| | Fundamentals of GPU Programming | Informatik | 5 | WS | benotet |
| | Information Theory | Informatik | 5 | SS | benotet |
| | Knowledge Graphs | Informatik | 5 | Letztmalig SS 23 | benotet |
| | Komplexitätstheorie | Informatik | 9 | WS | benotet |
| | Künstliche Neuronale Netze | Informatik | 6 | Letztmalig WS 22/23 | benotet |
| | Machine Learning: Supervised Methods | Informatik | 6 | SS | benotet |
| | Nebenläufige Programmierung | Informatik | 5 | SS | benotet |
| | Quantum Information and Computation | Informatik | 5 | Letztmalig WS 22/23 | benotet |
| | Web-Engineering | BauIng | 5 | Letztmalig SS 23 | benotet |

## Angebotene Vertiefungsseminare im Modul „Vertiefungsseminar (M.Sc. IT-Sicherheit)“

| | Lehrveranstaltung | Einheit | Umfang Modul (LP) | Semester | Bewertung |
|---|---|---|---|---|---|
| Vertiefungsseminare | | | | | |
| | Seminar zur Real World Cryptoanalysis | Informatik | 3 | SS und WS | unbenotet |
| | Seminar Human Centered Security and Privacy | Informatik | 3 | WS | unbenotet |
| | Seminar Information Security Seminar | Informatik | 3 | WS | unbenotet |
| | Master Seminar Security and Privacy for Mobile Systems | Informatik | 3 | SS und WS | unbenotet |
| | Master-Seminar "Digitale Souveränität" | Informatik | 3 | SS und WS | unbenotet |
| | Seminar Netz- und Datensicherheit | Informatik | 3 | SS und WS | unbenotet |
| | Seminar on Current Topics for Systems Security and Privacy | Informatik | 3 | SS und WS | unbenotet |
| | Seminar Quantum Cryptography | Informatik | 3 | WS | unbenotet |
| | Seminar Ressourceneffiziente Systemsoftwarekonzepte | Informatik | 3 | SS und WS | unbenotet |
| | Seminar Security Engineering | Informatik | 3 | SS und WS | unbenotet |
| | Seminar Software and Internet Security | Informatik | 3 | SS und WS | unbenotet |
| | Seminar zur symmetrischen Kryptographie | Informatik | 3 | WS | unbenotet |
| | Master-Seminar Developer Centered Security | Informatik | 3 | SS | unbenotet |
| | Seminar Implementation Security | Informatik | 3 | SS | unbenotet |
| | Seminar Quantum Algorithms | Informatik | 3 | SS | unbenotet |
| | Seminar Satisfiability | Informatik | 3 | SS | unbenotet |
| | Seminar zur symmetrischen Kryptographie | Informatik | 3 | SS | unbenotet |
| | Information Security Seminar Bachelor | Informatik | 3 | SS | unbenotet |

## Angebotene Praktika im Modul „Master Praktikum/Projektarbeit IT-Sicherheit“

| | Lehrveranstaltung | Einheit | Umfang Modul (LP) | Semester | Bewertung |
|---|---|---|---|---|---|
| | Projekt Netz- und Datensicherheit | Informatik | 4 | SS und WS | unbenotet |
| | Forschungspraktikum Human-Centred Security | Informatik | 4 | WS | unbenotet |
| | Praktikum TLS Implementierung | Informatik | 4 | WS | unbenotet |
| | Praktikum zur Hackertechnik (Hackerpraktikum) | Informatik | 4 | WS | unbenotet |
| | Developer Centered Security | Informatik | 4 | WS | unbenotet |
| | Master-Forschungspraktikum (Laborstudien) Human-Centred Security | Informatik | 4 | WS | unbenotet |
| | Master-Praktikum Wireless Physical Layer Security | Informatik | 4 | WS | unbenotet |
| | Research in Information Security (Master Project) | Informatik | 4 | SS und WS | unbenotet |
| | Master-Praktikum Reverse-Engineering Security Features | 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 |

Abkürzungen:

SS: Sommersemester
WS: Wintersemester

CP: Creditpoints

ETIT: Fakultät für Elektrotechnik und Informationstechnik
BauIng: 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

Fundamentals of GPU Programming

Information Theory

Komplexitätstheorie

Machine Learning: Supervised Methods

Nebenläufige Programmierung

Aktuelle Themen im Bereich der Internet-Sicherheit

Aufbau eines Managementsystems für Informationssicherheit nach DIN ISO / 27001

Blockchain Security and Privacy

Developer Centered Security

Empirische IT-Sicherheitsforschung

Human Aspects of Cryptography Adoption

Menschliches Verhalten in der IT-Sicherheit

Message Level Security

Microarchitectural Attacks and Defenses

Privacy, data governace and usability

Processor Security

Programmanalyse

Software Protection

Software Security

Software-Implementierung kryptographischer Verfahren

Authentische Schlüsselvereinbarung: Formale Modelle und Anwendungen

Foundations of Programming Languages, Verification, and Security

Fundamentals of Data Science

Kryptographische Protokolle

Proofs are programs

Public Key Kryptanalyse 1

Public Key Verschlüsselung (kein Angebot im WS 23/24)

Quantum Cryptography (kein Angebot im WS 23/24)

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 | Workload | Semester | Turnus | Dauer |
|---|---|---|---|---|---|
| | 9 CP | 270 h | siehe Prüfungsordnung / see Examination regulations | Wintersemester | 1 Semester |

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Advanced Algorithms (212029) | 90 h | 180 h | 40 Studierende |

| Unterrichtssprache | Teilnahmevoraussetzungen |
|---|---|
| Englisch | 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)**
- 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: Aktuelle Themen im Bereich der Internet-Sicherheit

| Modul-Nr./Code | Credits | Workload | Semester | Turnus | Dauer |
|---|---|---|---|---|---|
| | 5 CP | | | Wintersemester | 1 Semester |

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| 211099 - Aktuelle Themen im Bereich der Internet-Sicherheit | | | Studierende |

| Unterrichtssprache | Teilnahmevoraussetzungen |
|---|---|
| Deutsch | |

**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:

- Portable Document Flaws
- Overview over Cryptographic Modelling with the Example of Messaging
- 0-RTT and Tor
- Padding Oracles
- Racoon
- Breaking Microsoft RMS 2020
- IPsec-Bleichenbacher
- DEMONS: DNS-Poisoning by Exhaustive Misappropriation of Network Sockets
- DOM
- XS Leaks
- UI Redressing

Neben den Systemen selbst werden dabei auch publizierte Angriffe auf diese Systeme besprochen; die Studierenden stellen selbst wissenschaftliche Überlegungen zur Verbesserung der Sicherheit an.

**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 wissenschaftsnahe 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 | Workload | Semester | Turnus | Dauer |
|---|---|---|---|---|---|
| | 4 CP | 120 h | siehe Prüfungsordnung | Sommersemester | 1 Semester |

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Aufbau eines Managementsystems für Informationssicherheit nach DIN ISO / IEC 27001 (211021) | 45 h | 75 h | Studierende |

| Unterrichtssprache | Teilnahmevoraussetzungen |
|---|---|
| Deutsch | Keine |

**Modulbeauftragte/r und hauptamtlich Lehrende**
Modulbeauftragte/r: Professur für Systemsicherheit
Lehrende: Dr.-Ing. Sebastian Uellenbeck

**Verwendung des Moduls**
B.Sc. IT-Sicherheit/ Informationstechnik

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&2 und Netzsicherheit 1&2

**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. Studierenden 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:

- 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

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).

**Lehrformen**
Vorlesung mit Übung (Blockveranstaltung in den Semesterferien Anmeldung über syssec@rub.de)

| |
|---|
| **Prüfungsformen**<br>schriftliche Modulabschlussprüfung (90 Minuten) |
| **Voraussetzungen für die Vergabe von Credits**<br>bestandene schriftliche Modulabschlussprüfung |
| **Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**<br><br>4/150: B.Sc. IT-Sicherheit/ Informationstechnik [PO 22]<br><br>4/149: B.Sc. IT-Sicherheit/ Informationstechnik [PO 20]<br><br>4/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22]<br><br>5/99: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 20] |

## Titel des Moduls: Authentische Schlüsselvereinbarung: Formale Modelle und Anwendungen

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

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Au-then-ti-sche Schlüs-sel-ver-ein-ba-rung: For-ma-le Mo-del-le und An-wen-dun-gen (211038) | 60 h | 90 h | 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**

as 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 Widerholung der Wahrscheinlichkeitstehorie, 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 | Workload | Semester | Turnus | Dauer |
|---|---|---|---|---|---|
| | 5 CP | 150 h | siehe Prüfungsordnung / see examination regulations | Sommersemester | 1 Semester |

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Autonomous Vehicles and Artificial Intelligence (211044) | 60 h | 90 h | 25 Studierende |

| Unterrichtssprache | Teilnahmevoraussetzungen |
|---|---|
| Englisch | 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 FahrzeugenLokalisierung 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 | Workload | Semester | Turnus | Dauer |
|---|---|---|---|---|---|
| | 5 CP | 150h | siehe Prüfungsordnung / see examination regulations | Wintersemester | 1 Semester |

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Autonomous Vehicles and Artificial Intelligence Lab (212035) | 60h | 90h | Studierende |

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

**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:
<ul>
<li>Teilnahme an der Vorlesung Autonomous Vehicles and Artificial Intelligence (211044)</li>
<li>Programmiererfahrung in C++ oder Python (z.B. als Teil von anderen Kursen)</li>
<li>Teilnahme an der Vorlesung Software Engineering (212000) oder einer vergleichbaren Lehrveranstaltung</li>
</ul>

**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 Entwicklungsprozess 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 Security and Privacy
**Blockchain Security and Privacy**

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

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Blockchain security and privacy (212007) | 60 h | 90 h | 40 Studierende |

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

**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**
keine

**Lernziele (learning outcomes)**
Nach Abschluss dieses Kurses sollen die Teilnehmer in der Lage sein:
1. die Definitionen von Sicherheit und Datenschutz bei offenen Zahlungssystemen zu erklären.
2. die Sicherheit von PoW-Blockchains vor dem Hintergrund des aktuellen Stands der Technik und der gemeldeten Angriffe zu erläutern.
3. mögliche Netzwerksicherheits- und kryptografische Gegenmaßnahmen zur Abwehr von Angriffen auf Blockchains zu erläutern.
4. Erläuterung der besten Sicherheits-/Privatsphärenpraktiken zur Stärkung der Sicherheit bestehender Blockchains und Ableitung relevanter Lehren für die Entwicklung von Blockchain-Technologien der nächsten Generation.

**Inhalt**
Das Hauptziel des Kurses ist es, einen umfassenden Überblick über die Sicherheit und den Datenschutz von Blockchain-Technologien zu geben.

Die Kursteilnehmer werden auch in die grundlegenden Sicherheits- und Datenschutzbestimmungen bestehender gängiger Währungen eingeführt und mit den neuesten Angriffen und Bedrohungen vertraut gemacht, die gegen bestehende Systeme/Einführungen gemeldet wurden. Die Teilnehmer werden auch über die Wirksamkeit der Kombination von Sicherheitsprimitiven auf Netzwerkebene mit neuartigen kryptografischen Primitiven zur Abwehr von Angriffen auf Zahlungssysteme nachdenken.

**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 | Workload | Semester | Turnus | Dauer |
|---|---|---|---|---|---|
| | 5 CP | 150 h | siehe Prüfungsordnung / see examination regulations | Wintersemester | 1 Semester |

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Deep Learning (212018) | 60 h | 90 h | 50 Studierende |

| Unterrichtssprache | Teilnahmevoraussetzungen |
|---|---|
| Deutsch | |

**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 | Workload | Semester | Turnus | Dauer |
|---|---|---|---|---|---|
| | 5 CP | 150 h | siehe Prüfungsordnung / see examination regulations | Sommersemester | 1 Semester |

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Deterministic Network Caluculus (211054) | 60 h | 90 h | Studierende |

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

**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,

- 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:

- 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 (30 Minuten)

**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: Developer Centered Security
**Developer Centered Security**

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

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Developer Centered Security (211050) | 45 h | 105 h | Studierende |

| Unterrichtssprache | Teilnahmevoraussetzungen |
|---|---|
| Deutsch | keine |

**Modulbeauftragte/r und hauptamtlich Lehrende**
Modulbeauftragte/r: Jun.-Prof. Dr. Alena Naiakshina
Lehrende: Jun.-Prof. Dr. Alena Naiakshina

**Verwendung des Moduls**
M.Sc. IT-Sicherheit/ Informationstechnik

M.Sc. IT-Sicherheit/ Netze und Systeme

**Vorkenntnisse**
keine

**Lernziele (learning outcomes)**
Benutzbarkeitsprobleme, Sicherheitsanforderungen und Schwachstellen aktueller Systeme kennen. Methodik zur Untersuchung der Benutzbarkeit von Sicherheitsfunktionalitäten verstehen. Verhaltensstudien mit Softwareentwicklern und Administratoren unter Beachtung der vorgestellten Guidelines durchführen können. Sichere und benutzerfreundliche Systeme für Softwareentwickler und Administratoren entwickeln und beurteilen können.

**Inhalt**
Softwareentwickler und Administratoren sind häufig keine Sicherheitsexperten. Die von ihnen gebauten Systeme weisen daher oft Sicherheitslücken auf, durch die Millionen Nutzer und vertrauliche Daten gefährdet werden. Wie genau kommt es aber dazu, dass Softwareentwickler und Administratoren solche gravierenden Sicherheitsfehler machen, obwohl es fertige Anwendungsschnittstellen (application programming interface (API)), Programmbibliotheken und Tools gibt, die das Entwicklern und Verwenden von Sicherheitskonzepten erleichtern sollen? Es wird ein Einblick in die Grundlagen der benutzbaren Sicherheit und Privatsphäre sowie aktuelle, sicherheitsrelevante Studien mit Softwareentwicklern und Administratoren gegebenen. Die daraus gewonnenen Erkenntnisse werden systematisch aufgearbeitet und dargelegt. Es wird ferner aufgezeigt, was Sicherheitssystemdesigner, Toolentwickler, und Kryptographen beim Entwurf ihrer Systeme beachten sollten, um Softwareentwickler und Administratoren dabei zu unterstützen sicherheitskritische Fehler zu vermeiden. Zudem werden Guidelines zum Durchführen von Studien mit Softwareentwicklern und Administratoren vorgestellt. Dabei wird eine Abgrenzung zu Studien mit Endbenutzern gezogen.

**Lehrformen**
Vorlesung mit Übung

**Prüfungsformen**
Schriftliche Modulabschlussprüfung

**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: Effiziente Algorithmen

| Modul-Nr./Code | Credits | Workload | Semester | Turnus | Dauer |
|---|---|---|---|---|---|
| | 9 CP | 270 h | siehe Prüfungsordnung | Sommersemester | 1 Semester |

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Effiziente Algorithmen (150320 + 150321) | 90 h | 180 h | Studierende |

| Unterrichtssprache | Teilnahmevoraussetzungen |
|---|---|
| Deutsch | |

**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

- 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. Laufzeitder 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:

- 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 | Workload | Semester | Turnus | Dauer |
|---|---|---|---|---|---|
| | 6 CP | 180 h | | Sommersemester | 1 Semester |

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Embedded Multimedia | 60 h | 120 h | Studierende |

| Unterrichtssprache | Teilnahmevoraussetzungen |
|---|---|
| Deutsch | 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 fur 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 Durchfuhrung 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 plattformubergreifenden 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 Ubungen

**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
**Empirical Security Research**

| Modul-Nr./Code | Credits | Workload | Semester | Turnus | Dauer |
|---|---|---|---|---|---|
| | 5 CP | 150 h | siehe Prüfungsordnung | Wintersemester | 1 Semester |

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Empirische IT-Sicherheitsforschung (212036) | 60 h | 90 h | Studierende |

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

**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]

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

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

| Modul-Nr./Code | Credits | Workload | Semester | Turnus | Dauer |
|---|---|---|---|---|---|
| | 6 CP | 180 h | siehe Prüfungsordnung / see examination regulations | Wintersemester | 1 Semester |

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Energy-Aware Computing Systems (212030) | 60 h | 120 h | 20 Studierende |

| Unterrichtssprache | Teilnahmevoraussetzungen |
|---|---|
| Englisch | 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: Foundations of Programming Languages, Verification, and Security
**Foundations of Programming Languages, Verification, and Security**

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

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Foundations of Programming Languages, Verification, and Security (211044) | 60 h | 90 h | 20 Studierende |

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

**Modulbeauftragte/r und hauptamtlich Lehrende**
Modulbeauftragte/r: Dr. Roberto Blanco
Dr. Catalin Hritcu
Lehrende: Dr. Roberto Blanco
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 having a working knowledge of the contents of the Logical Foundations book (<a href="https://mpi-sp-pap-2023.github.io/book">https://mpi-sp-pap-2023.github.io/book</a>), 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

l understand how to define in Coq the syntax of simple programming languages: variants of a simple imperative language and the simply-typed lambda calculus;

l define the big-step and small-step operational semantics of such simple languages;

l formally define type systems for such languages as inductive relations;

l work out the metatheory of such languages, by proving results such as type soundness;

l understand the semantic foundations of Hoare Logic and Relational Hoare Logic;

l use Hoare Logic for verifying the correctness of simple imperative programs, both formally in Coq and informally on paper;

l understand the semantic foundations of Secure Information Flow Control and Noninterference.

l use Relational Hoare Logic for proving program equivalence as well as Noninterference of simple imperative programs;

**Inhalt**
Complex proofs on paper are difficult to write, 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 mathematical results.

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 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 and Cryptographic Constant Time as well as their formal noninterference guarantees.

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

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

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

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

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

## Titel des Moduls: Fundamentals of Data Science
**Fundamentals of Data Science**

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

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Fundamentals of Data Science (141213) | 60 h | 90 h | Studierende |

| Unterrichtssprache | Teilnahmevoraussetzungen |
|---|---|
| Deutsch | |

**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**
<ul>
<li>Mathematik I-IV</li>
<li>Systemtheorie I-III</li>
<li>Optimierung</li>
</ul>

**Lernziele (learning outcomes)**

**Inhalt**
Die Modulnote setzt sich aus zwei Anteilen zusammen: 1.Note der muendlichen Pruefung (36 %) 2. Note der Ausarbeitung eines wissenschaftlichen Artikels und des dazugehoerigen 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 | Workload | Semester | Turnus | Dauer |
|---|---|---|---|---|---|
| | 5 CP | 150 h | siehe Prüfungsordnung | Wintersemester | 1 Semester |

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Fundamentals of GPU Programming (141374) | 45 h | 105 h | Studierende |

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

**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 Rechenkerne. 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**
Semesterbegleitend: Projektarbeit und schriftliche Prüfung

**Voraussetzungen für die Vergabe von Credits**
Bestehen der Projektarbeit und der schriftlichen Prüfung

**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: Human Aspects of Cryptography Adoption
**Human Aspects of Cryptography Adoption**

| Modul-Nr./Code | Credits | Workload | Semester | Turnus | Dauer |
|---|---|---|---|---|---|
| | 5 CP | 150 h | siehe Prüfungsordnung | Wintersemester | 1 Semester |

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Human Aspects of Cryptography Adoption | 60 h | 90 h | 30 Studierende |

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

1. cryptographic solutions – which experts agree offer good protection against most oft 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.

- 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

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

## Titel des Moduls: Information Theory
**Information Theory**

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

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Information Theory (211007) | 60 h | 90 h | 30 Studierende |

| Unterrichtssprache | Teilnahmevoraussetzungen |
|---|---|
| Englisch | 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: Komplexitätstheorie
**Computational complexity theory**

| Modul-Nr./Code | Credits | Workload | Semester | Turnus | Dauer |
|---|---|---|---|---|---|
| | 9 CP | 270 h | siehe Prüfungsordnung / see examination regulations | Wintersemester | 1 Semester |

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Computational complexity theory (211028) | 90 h | 180 h | Studierende |

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

**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**<br>Mündliche Modulabschlussprüfung (20-30 Minuten) |
| **Voraussetzungen für die Vergabe von Credits**<br>Bestandene mündliche Modulabschlussprüfung |
| **Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**<br><br>9/97: M.Sc. Computer Science<br><br>9/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22]<br><br>9/84: M.Sc. IT-Sicherheit/ Informationstechnik [PO 20]<br><br>9/99: M.Sc. IT-Sicherheit/ Netze und Systeme [PO22]<br><br>9/96: M.Sc. IT-Sicherheit/ Netze und Systeme [PO20] |

## Titel des Moduls: Kryptographische Protokolle
**Cryptographic protocols**

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

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Kryptographische Protokolle (211031) | 60 h | 90 h | Studierende |

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

**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)**
- 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:
- 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
**Machine Learning: Supervised Methods**

| Modul-Nr./Code | Credits | Workload | Semester | Turnus | Dauer |
|---|---|---|---|---|---|
| | 6 CP | 180 h | siehe Prüfungsordnung / see examination regulations | Sommersemester | 1 Semester |

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Machine Learning: Supervised Methods (211024) | 60 h | 120 h | 80 Studierende |

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

**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 Vidoes und Lesematerial vermittelt. Übungsaufgaben mit Programmieranteilen werden in Form von Jupyter-Notebooks bereitgestellt. Nach dem erfolgreichen Abschluss des Moduls

- 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**

Bestandene schriftliche Modulabschlussprüfung

**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 | Workload | Semester | Turnus | Dauer |
|---|---|---|---|---|---|
| | 4 CP | 120 h | 3 | jedes Semester | 1 Semester |

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| • Praktische Kryptanalyse von symmetrischen Chiffren (211401)<br>• Projekt Netz- und Datensicherheit (212412)<br>• Forschungspraktikum Human-Centred Security (212408)<br>• Initial Research in Information Security (212402)<br>• Praktikum TLS Implementierung (212414)<br>• Praktikum zur Hackertechnik (Hackerpraktikum) (212413)<br>• Research in Software/Internet Security (2124)<br>• Master-Praktikum ARM Processors for Embedded Cryptography (212407)<br>• Developer Centered Security (212417)<br>• Praktikum Wireless Physical Layer Security (142025) | je nach Veranstaltungswahl | abhängig von der Praktikumswahl | Studierende |

| Unterrichtssprache | Teilnahmevoraussetzungen |
|---|---|
| abhängig von der Praktikumswahl: Deutsch oder Englisch | |

**Modulbeauftragte/r und hauptamtlich Lehrende**
Modulbeauftragte/r: Studiendekan IT-Sicherheit
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

- haben Studierende Ihre Fähigkeiten in der Analyse und dem Einsatz von Verfahren zur Sicherung von IT-Systemen vertieft und erweitert
- je nach gewählten Praktikum können noch weitere Lernziele dazu kommen

**Inhalt**

Es werden in jedem Semester einige Praktika aus folgendem Katalog angeboten:

- Praktische Kryptanalyse von symmetrischen Chiffren
- Projekt Netz- und Datensicherheit
- Forschungspraktikum Human-Centred Security
- Laborstudien Human-Centred Security
- Initial Research in Information Security
- Praktikum TLS Implementierung
- Praktikum zur Hackertechnik (Hackerpraktikum)
- Research in Software/Internet Security

- Master-Praktikum ARM Processors for Embedded Cryptography
- Developer Centered Security (Projekt)
- Praktikum Wireless Physical Layer Security

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 | Workload | Semester | Turnus | Dauer |
|---|---|---|---|---|---|
| | 5 CP | 150 h | siehe Prüfungsordnung | Sommersemester | 1 Semester |

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Menschlisches Verhalten in der IT-Sicherheit (211033) | 60 h | 90 h | Studierende |

| Unterrichtssprache | Teilnahmevoraussetzungen |
|---|---|
| Deutsch | 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 uber Forschungs- ¨ methoden im Bereich usable Security mit einem besonderen Schwerpunkt auf Laborstudien. Es werden theoretische Kenntnisse vermittelt, auf deren Grundlage die Studierenden selbstst¨andig eine Laborstudie planen und umsetzen und auf diese Weise praktische Kenntnisse erwerben sollen.

**Inhalt**
In *Menschliches Verhalten in der IT-Sicherheit* 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]

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

## Titel des Moduls: Message Level Security
**Message Level Security**

| Modul-Nr./Code | Credits | Workload | Semester | Turnus | Dauer |
|---|---|---|---|---|---|
| | 5 CP | 150 h | siehe Prüfungsordnung | Wintersemester | 1 Semester |

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Message-Level Security (212060) | 60 h | 90 h | Studierende |

| Unterrichtssprache | Teilnahmevoraussetzungen |
|---|---|
| Deutsch | |

**Modulbeauftragte/r und hauptamtlich Lehrende**
Modulbeauftragte/r: Dr.-Ing. Christan Mainka
Lehrende: Dr.-Ing. Christan Mainka
Dr.-Ing. Vladislav Mladenov

**Verwendung des Moduls**
B.Sc. IT-Sicherheit/ Informationstechnik

M.Sc. IT-Sicherheit/ Informationstechnik

M.Sc. IT-Sicherheit/ Netze und Systeme

**Vorkenntnisse**
Vorlesung Netzsicherheit 2<br/>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:

- **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 verbreitetste 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)**

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

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

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 | Workload | Semester | Turnus | Dauer |
|---|---|---|---|---|---|
| | 5 CP | 150 h | siehe Prüfungsordnung / see examination regulations | Wintersemester | 1 Semester |

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Microarchitectural Attacks and Defenses (212064) | 45 h | 105 h | 30 Studierende |

| Unterrichtssprache | Teilnahmevoraussetzungen |
|---|---|
| Englisch | 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)**
- 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: Nebenläufige Programmierung

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

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Nebenläufige Programmierung (211012) | 45 h | 105 h | Studierende |

| Unterrichtssprache | Teilnahmevoraussetzungen |
|---|---|
| Deutsch | keine |

**Modulbeauftragte/r und hauptamtlich Lehrende**
Modulbeauftragte/r: Dr.-Ing. Doga Arinir (Lehrauftrag)
Lehrende: Dr.-Ing. Doga Arinir

**Verwendung des Moduls**
B.Sc. Informatik

B.Sc. Angewandte Informatik

Master IT-Sicherheit/ Informationstechnik

**Vorkenntnisse**
Beherrschung einer Objektorientierten Programmiersprache (idealerweise Java)

**Lernziele (learning outcomes)**
Nach dem erfolgreichen Abschluss des Moduls

- haben die Studierenden grundlegende Fähigkeiten und Techniken erworben, um nebenläufige Programme sicher entwickeln zu können
- kennen die Studierenden softwaretechnische Entwurfsmuster, welche bekannte Probleme bei nebenläufigen Programmen, wie zum Beispiel die Verklemmung, vermeiden lassen
- können die Studierenden die Performanz von Programmen durch den Einsatz der nebenläufigen Programmierung verbessern
- sind die Studierenden in der Lage, bestehende Programme zu analysieren und mögliche Fehler zu erkennen
- können die Studierenden die Sprachmerkmale und Schnittstellen von JAVA für die nebenläufige Programmierung sicher anwenden

**Inhalt**
Moderne Hardware-Architekturen lassen sich nur durch den Einsatz nebenläufiger Programme richtig ausnutzen. Die nebenläufige Programmierung garantiert bei richtiger Anwendung eine optimale Auslastung der Hardware. Jedoch sind mit einem sorglosen Einsatz dieser Technik auch viele Risiken verbunden. Die Veranstaltung stellt Vorteile und auch Probleme nebenläufiger Programme dar und zeigt, wie sich die Performanz von Programmen verbessern lässt.

1. Nebenläufigkeit: Schnelleinstieg

- Anwendungen vs. Prozesse
- Programme und ihre Ausführung
- Vorteile und Probleme von nebenläufigen Programmen (Verbesserung der Perfor-manz, Synchronisation, Realisierung kritischer Abschnitte, Monitore, Lebendigkeit, Verklemmungen)

2. Threads in Java

3. UML-Modellierung von Nebenläufigkeit

4. Neues zur Nebenläufigkeit in Java 5 und Java 6

5. Realisierung von Nebenläufigkeit 6. Fortschritte Java-Konzepte für Nebenläufigkeit

6. Fortschritte Java-Konzepte für Nebenläufigkeit

**Lehrformen**

Online Vorlesung mit begleitendem eLearning Kurs

**Prüfungsformen**
Schriftliche Modulabschlussprüfung über 90 Minuten

**Voraussetzungen für die Vergabe von Credits**
Bestandene Modulabschlussprüfung

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

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

5/165: B.Sc. Informatik [PO 20]

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

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

5/91 Master IT-Sicherheit/ Informationstechnik

## Titel des Moduls: Privacy, data governace and usability

**Privacy, data governace and usability**

| Modul-Nr./Code | Credits | Workload | Semester | Turnus | Dauer |
|---|---|---|---|---|---|
| | 5 CP | 150 h | siehe Prüfungsordnung | Wintersemester | 1 Semester |

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Privacy, data governace and usability (212037) | 45 h | 105 h | 20 Studierende |

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

**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: <br/> • Einführung in die Usable Security and Privacy (211036)<br/> • Datenschutz (260081)<br/> • Basic knowledge of threat modeling <br/> • 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**

**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: Processor Security
**Processor Security**

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

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Processor Security (211099) | 60 h | 90 h | Studierende |

| Unterrichtssprache | Teilnahmevoraussetzungen |
|---|---|
| Deutsch | |

**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-CPUs, 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

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

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Programmanalyse (211015) | 60 h | | Studierende |

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

**Vorkenntnisse**
keine

**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:

- 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 Semester 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]

## Titel des Moduls: Proofs are programs
**Proofs are programs**

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

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Proofs are Programms (211003) | 60 h | 90 h | 40 Studierende |

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

**Modulbeauftragte/r und hauptamtlich Lehrende**
Modulbeauftragte/r: Dr. Catalin Hritcu
Lehrende: Dr. Catalin Hritcu
Dr. Clara Schneidewind

**Verwendung des Moduls**
B.Sc. Informatik

B.Sc. IT-Sicherheit/ Informationstechnik

M.Sc. IT-Sicherheit/ Informationstechnik

M.Sc. IT-Sicherheit/ Netze und Systeme

**Vorkenntnisse**

**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, structural induction, proof automation);
- define new inductive types and relations in Coq and prove statements about them;
- 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;
- understand 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.

**Inhalt**
Complex mathematical proofs on paper are difficult to write, 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 mathematical results such as the four color theorem, the odd-order theorem (Feit–Thompson), or the construction of perfectoid spaces.

This course introduces the Coq proof assistant and explains how to use it to prove properties about functional programs and inductive relations. 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. This deep connection between programs and proofs should make this course interesting to both computer scientists and mathematicians. For computer

scientists the goal is to demystify proofs as just programs in an elegant programming language, for which the course provides a gentle introduction. For mathematicians this course serves as an introduction to functional programming and also to the idea 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 online textbook, 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. Finally, this course serves as the base for a more advanced course on “Foundations of Programming Languages, Verification, and Security”.

**Lehrformen**

**Prüfungsformen**

**Voraussetzungen für die Vergabe von Credits**

**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/ Informationstechnik [PO 22]

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

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: Public Key Kryptanalyse 1
**Public Key Cryptanalysis 1**

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

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Public Key Kryptanalyse 1 (211055) | 45 h | 105 h | Studierende |

| Unterrichtssprache | Teilnahmevoraussetzungen |
|---|---|
| Deutsch | keine |

**Modulbeauftragte/r und hauptamtlich Lehrende**
Modulbeauftragte/r: Prof. Alex May
Lehrende: Prof. Alex May

**Verwendung des Moduls**

**Vorkenntnisse**
Vorausgesetzt werden elementare Kenntnisse der Lineare Algebra (Mathematik 1 für 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

- 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,
- sind Studierende in der Lage, Techniken der Kryptanalyse mit Hilfe der Computer-Algebra Sage zu implementieren.

**Inhalt**
Kryptanalyse dient dazu, kryptographische Systeme derart zu instantiieren, 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**

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

5/149 B.Sc. IT-Sicherheit [PO20]

5/150 B.Sc. IT-Sicherheit [PO22]

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

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

## Titel des Moduls: Public Key Verschlüsselung (kein Angebot im WS 23/24)

| Modul-Nr./Code | Credits | Workload | Semester | Turnus | Dauer |
|---|---|---|---|---|---|
| | 5 CP | 150 h | siehe Prüfungsordnung | Wintersemester | 1 Semester |

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Public Key Verschlüsselung | 60 h | 90 h | Studierende |

| Unterrichtssprache | Teilnahmevoraussetzungen |
|---|---|
| Deutsch | |

**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 fur 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 Verschlusselung. 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 23/24)
**Quantum Cryptography**

| Modul-Nr./Code | Credits | Workload | Semester | Turnus | Dauer |
|---|---|---|---|---|---|
| | 5 CP | 150 h | siehe Prüfungsordnung | Wintersemester | 1 Semester |

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Quantum Cryptography (212016) | 60 h | 90 h | Studierende |

| Unterrichtssprache | Teilnahmevoraussetzungen |
|---|---|
| Englisch | 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: Software Protection
**Software Protection**

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

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Software Protection (211107) | 45 h | 105 h | Studierende |

| Unterrichtssprache | Teilnahmevoraussetzungen |
|---|---|
| Deutsch | keine |

**Modulbeauftragte/r und hauptamtlich Lehrende**
Modulbeauftragte/r: Studiendekan IT-Sicherheit
Lehrende: Dr.-Ing. Tim Blazytko
Philipp Koppe

**Verwendung des Moduls**
B.Sc. IT-Sicheheit/ Informationstechnik

M.Sc. IT-Sicherheit/ Informationstechnik

M.Sc. IT-Sicherheit/ Netze und Systeme

**Vorkenntnisse**
Im Bereich Reverse Engineering sind empfohlen, beipielsweise 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 Obfuskierungstechniken 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, verschiede 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 Ausfuehrung

- SMT Solving

- Programmsynthese

- Überblick zu existierenden Analysetools und Frameworks

**Lehrformen**
Vorlesung mit Übung

**Prüfungsformen**
Arbeit/Kompetenznachweis im Semester. Die Lehrveranstaltung beinhaltet mehere 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-Sicheheit/ Informationstechnik [PO 22]

5/149: Bachelor IT-Sicheheit/ 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
**Software Security**

| Modul-Nr./Code | Credits | Workload | Semester | Turnus | Dauer |
|---|---|---|---|---|---|
| | 9 CP | 270 h | siehe Prüfungsordnung | Wintersemester | 1 Semester |

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Software Security (212026) | 60 h | 210 h | Studierende |

| Unterrichtssprache | Teilnahmevoraussetzungen |
|---|---|
| Englisch | Systemsicherheit und Betriebssysteme |

**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

**Vorkenntnisse**
Keine

**Lernziele (learning outcomes)**

**Inhalt**

**Lehrformen**
Vorlesung mit Übung

**Prüfungsformen**
Schriftliche Hausarbeit (Take-Home-Exam) am Ende der Vorlesungszeit.

**Voraussetzungen für die Vergabe von Credits**

Bestandene Hausarbeit.

**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

## Titel des Moduls: Software-Implementierung kryptographischer Verfahren

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

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Software-Implementierung kryptographischer Verfahren (211035) | 60 h | 90 h | Studierende |

| Unterrichtssprache | Teilnahmevoraussetzungen |
|---|---|
| Deutsch | keine |

**Modulbeauftragte/r und hauptamtlich Lehrende**
Modulbeauftragte/r: Prof. Dr.-Ing. Tim Guneysu
Lehrende: Dr.-Ing. Max Hoffmann

**Verwendung des Moduls**
B.Sc. IT-Sicherheit/ Informationstechnik

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:

- 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)

**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)**

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

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

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 | Workload | Semester | Turnus | Dauer |
|---|---|---|---|---|---|
| | 5 CP | 150 h | siehe Prüfungsordnung | Wintersemester | 1 Semester |

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Symmetrische Kryptanalyse | 60 h | 90 h | Studierende |

| Unterrichtssprache | Teilnahmevoraussetzungen |
|---|---|
| Deutsch | |

**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

M.Sc. Angewandte Informatik

**Vorkenntnisse**
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) Attacken, Impossible Differentials, Boomerang-Angriffe und Slide-Attacken. 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]

5/105: M.Sc. Angewandte Informatik

## Titel des Moduls: Vertiefungsseminar (M.Sc. IT-Sicherheit)

| Modul-Nr./Code | Credits | Workload | Semester | Turnus | Dauer |
|---|---|---|---|---|---|
| | 3 CP | 90 h | | jedes Semester | Semester |

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| 211104 Human Centred Security and Privacy<br>211110 Seminar Real-World Kryptanalyse<br>211114 Master-Seminar on Security and Privacy of Mobile Operating Systems<br>211117 Seminar Satisfiability (bis SoSe 23)<br>211119 Quantum Algorithms (bis SoSe 23)<br>211121 Fortgeschrittene Themen des Model Checking ( )<br>211122 Seminar über Grenzen in der theoretischen Informatik ( )<br>211129 Master-Seminar Developer Centered Security<br>211132 Master-Seminar Digitale Souveränität<br>211133 Seminar on Current Topics for Systems Security and Privacy<br>212109 Information Security Seminar<br>212111 Seminar Ressourceneffiziente Systemsoftware<br>212112 Seminar Security Engineering<br>212118 Seminar zur symmetrischen Kryptographie<br>212121 Seminar Netz- und Datensicherheit<br>212122 Seminar Current Topics in Device Firmware Security<br>212125 Software and Internet Security Seminar<br>212126 Seminar Implementation Security (bis SoSe 23) | 30 h | | Studierende |

| Unterrichtssprache | Teilnahmevoraussetzungen |
|---|---|
| Deutsch oder Englisch | |

**Modulbeauftragte/r und hauptamtlich Lehrende**
Modulbeauftragte/r: Studiendekan IT-Sicherheit
Lehrende: siehe jeweiliges Seminar

**Verwendung des Moduls**

M.Sc. IT-Sicherheit / Informationstechnik

M.Sc. IT-Sicherheit / Netze und Systeme

**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

- 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

**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 | Workload | Semester | Turnus | Dauer |
|---|---|---|---|---|---|
| | 5 CP | 150 h | siehe Prüfungsordnung | Sommersemester | 1 Semester |

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| Ze-ro-Know-ledge Proof Sys-tems (211032) | 60 h | 90 h | Studierende |

| Unterrichtssprache | Teilnahmevoraussetzungen |
|---|---|
| Deutsch | 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**
Written Exam / Oral Exam
The form of examination will be determined at the beginning of the lecture.

**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]

## Titel des Moduls: Freie Wahlmodule
**free electives**

| Modul-Nr./Code | Credits | Workload | Semester | Turnus | Dauer |
|---|---|---|---|---|---|
| | 25 CP | | | | Semester |

| Lehrveranstaltungen | Kontaktzeit | Selbststudium | Gruppengröße |
|---|---|---|---|
| | siehe Lehrveranstaltungen | | 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:

- 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]