

MODULHANDBUCH

Übersicht der Module

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

Wahlpflichtbereich

Aktuelle Themen im Bereich der Internet-Sicherheit

Aufbau eines Managementsystems für Informationssicherheit nach DIN ISO / 27001

Developer Centered Security

Human Aspects of Cryptography Adoption

Menschliches Verhalten in der IT-Sicherheit

Message Level Security

Microarchitectural Attacks and Defenses

Processor Security

Programmanalyse

Software Protection

Software Security

Software-Implementierung kryptographischer Verfahren

Authentische Schlüsselvereinbarung: Formale Modelle und Anwendungen

Deep Learning

Fundamentals of Data Science

Komplexitätstheorie

Kryptographische Protokolle

Proofs are programs

Public Key Kryptanalyse 1

Public Key Verschlüsselung

Quantum Cryptography

Symmetrische Kryptanalyse

Zero-Knowledge Proof Systems

Autonomous Vehicles and Artificial Intelligence

Autonomous Vehicles and Artificial Intelligence Lab

Deterministic Network Calculus

Effiziente Algorithmen

Embedded Multimedia

Foundations of Programming Languages, Verification, and Security

Fundamentals of GPU Programming

Information Theory

Künstliche Neuronale Netze

Machine Learning: Supervised Methods

Nebenläufige Programmierung

Master Praktikum/Projektarbeit IT-Sicherheit
Vertiefungsseminar (M.Sc. IT-Sicherheit)

Wahlbereich

Freie Wahlfächer

Abschlussarbeit

Masterarbeit und Kolloquium (ITS)

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

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

Modul-Nr./Code	Credits 4 CP	Workload 120 h	Semester siehe Prüfungsordnung	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Aufbau eines Managementsystems für Informationssicherheit nach DIN ISO / IEC 27001 (211021)			Kontaktzeit 45 h	Selbststudium 75 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen Keine		

Modulbeauftragte/r und hauptamtlich Lehrende

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

Verwendung des Moduls

B.Sc. IT-Sicherheit/ Informationstechnik

M.Sc. IT-Sicherheit/ Informationstechnik

M.Sc. IT-Sicherheit/ Netze und Systeme

Vorkenntnisse

Vor-;kennt-;nis-;se über Sys-;tem-;si-;cher-;heit und Netz-;si-;cher-;heit z. B. aus den Vor-;le-;sun-;gen Sys-;tem-;si-;cher-;heit 1&2 und Netz-;si-;cher-;heit 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 sysec@rub.de)

Prüfungsformen

schriftliche Modulabschlussprüfung (90 Minuten)

Voraussetzungen für die Vergabe von Credits

bestandene schriftliche Modulabschlussprüfung

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

4/150: B.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

4/149: B.Sc. IT-Sicherheit/ Informationstechnik [PO 20]

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

5/99: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 20]

Titel des Moduls: Authentische Schlüsselvereinbarung: Formale Modelle und Anwendungen					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Au-then-ti-sche Schlüs-sel-ver-ein-ba-rung: For-ma-le Mo-del-le und An-wen-dun-gen (211038)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen Keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Jörg Schwenk Lehrende: Prof. Dr. Jörg Schwenk					
Verwendung des Moduls M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Netze und Systeme					
Vorkenntnisse <p>• Grundkenntnisse Kryptographie</p> <p>• Empfehlung: Durcharbeiten der ersten 40 Folien vom Skript Kryptographie I von Prof. Alexander May</p>					
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 Wiederholung der Wahrscheinlichkeitstheorie, Informationstheorie, etc.) • Beweisbare Sicherheit • Analyse von Schlüsselaustauschprotokollen, mit besonderem Fokus auf praktische Beispielprotokolle (wie TLS oder SSH) Die Zusammenstellung ist nicht fest und kann nach Absprache mit den Hörern auch geändert werden.					
Lehrformen Vorlesung mit Übung					
Prüfungsformen schriftlich, 120 Minuten					
Voraussetzungen für die Vergabe von Credits Bestandene Modulabschlussprüfung					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 5/91: M.Sc. IT-Sicherheit/ Informationstechnik					

Titel des Moduls: Autonomous Vehicles and Artificial Intelligence					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Autonomous Vehicles and Artificial Intelligence (211044)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße 25 Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Thorsten Berger Lehrende: Prof. Dr. Thorsten Berger, Dr. Sven Peldszus					
Verwendung des Moduls B.Sc. Informatik B.Sc. IT-Sicherheit B.Sc. Angewandte Informatik M.Sc. Informatik M.Sc. Angewandte Informatik M.Sc. IT-Sicherheit/Informationstechnik					
Vorkenntnisse <p>The Software Engineering lecture or a comparable course, Programming experiences e.g. as part of other courses</p>					
Lernziele (learning outcomes) <ul style="list-style-type: none"> • Understanding requirements on autonomous vehicles • Understanding the architecture of autonomous vehicles • Ability to build a self-driving car with ROS2 • Understanding and applying quality assurance for autonomous vehicles 					
Inhalt Autonomous driving is the future of individual mobility and all major manufacturers are working on fully autonomous vehicles. While there are robust and good solutions for the individual problems in autonomous driving, the main challenge lies in their integration. Altogether, an autonomous vehicle's software is the biggest problem. Therefore, the key in self-driving vehicles is about getting the software right. In this course, we will investigate the different aspects of self-driving vehicles as well as the importance and application of artificial intelligence in this domain. The course will primarily focus on the following topics: <ul style="list-style-type: none"> • Requirements on autonomous vehicles • Architecture of autonomous vehicles • Operation systems and frameworks for robotic systems • Specification and Implementation of autonomous vehicles based on ROS2 • Artificial intelligence for autonomous vehicles • Simulation of autonomous vehicles &#8729; Localization and perception • Mission planning • Quality assurance for autonomous vehicles In the course's lecture, we provide the required theoretical background and practically apply the course's content in exercises by building a self-driving robot. 					
Lehrformen Vorlesung mit Übungen					

Prüfungsformen

Mündliche Prüfung

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)

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

5/165: B.Sc. Informatik [PO 22]

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

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

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

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

5/ 97: M.Sc. Informatik

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					
Modul-Nr./Code	Credits 5 CP	Workload 150h	Semester siehe Prüfungsordnung	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Autonomous Vehicles and Artificial Intelligence Lab (212035)			Kontaktzeit 60h	Selbststudium 90h	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr Thorsten Berger Lehrende: Dr. Sven Peldszus					
Verwendung des Moduls					
Vorkenntnisse <p>Recommended: Autonomous Vehicles and Artificial Intelligence (Lecture); Programming experience in Python or C++; Software Engineering (Lecture)</p>					
Lernziele (learning outcomes)					
Inhalt Autonomous driving is the future of individual mobility and all major manufacturers are working on fully autonomous vehicles. While there are robust and good solutions for the individual problems in autonomous driving, the main challenge lies in their integration. Altogether, an autonomous vehicle's software is the biggest problem. Therefore, the key in self-driving vehicles is about getting the software right. In this course, we will practically explore the different aspects of self-driving vehicles as well as the importance and application of artificial intelligence in this domain by developing a self-driving race car. The course will primarily focus on the following topics: <ul style="list-style-type: none">• Requirements on autonomous vehicles• Architecture of autonomous vehicles• Operation systems and frameworks for robotic systems• Specification and Implementation of autonomous vehicles based on ROS2• Artificial intelligence for autonomous vehicles• Simulation of autonomous vehicles• Localization & perception• Mission planning• Quality assurance for autonomous vehicles					
Lehrformen Lecture with practical exercise					

Prüfungsformen
Voraussetzungen für die Vergabe von Credits
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

Titel des Moduls: Deep Learning					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Deep Learning (212018)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Asja Fischer Lehrende: Prof. Dr. Asja Fischer					
Verwendung des Moduls M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Netze und Systeme M.Sc. Angewandte Informatik M.Sc. Informatik					
Vorkenntnisse					
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] 5/99: 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 5/ 97: M.Sc. Informatik					

Titel des Moduls: Deterministic Network Calculus Deterministic Network Calculus					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Deterministic Network Calculus (211054)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Steffen Bondorf Lehrende: Prof. Dr. Steffen Bondorf					
Verwendung des Moduls M.Sc. Informatik M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. Angewandte Informatik					
Vorkenntnisse <p>Mathematik (Funktionsanalyse), Computernetze / Verteilte Systeme</p>					
Lernziele (learning outcomes) Nach erfolgreichem Abschluss des Moduls werden die Studierenden in der Lage sein, <ul style="list-style-type: none"> • komplexe, vernetzte Systeme als deterministische Warteschlangensysteme zu modellieren, • worst-case Leistungsanalysen von bestehenden Systemen bzw. Modellen durchzuführen, • die Herausforderungen bei der Leistungsdimensionierung von geplanten Systemen zu verstehen, &#8729; dabei die Wirkungsweise zentraler Mechanismen in Computernetzen anhand des Network Calculus zu erklären, • die vorgestellten Verfahren gegeneinander abzugrenzen und auf wissenschaftliche Fragestellungen anzuwenden. 					
Inhalt Verteilte Systeme sind heutzutage allgegenwärtig, und ihre Vernetzung ist von grundlegender Bedeutung für die kontinuierliche Verbreitung und damit Verfügbarkeit von Daten. Die Bereitstellung von Daten in Echtzeit ist einer der wichtigsten nichtfunktionalen Aspekte, den sicherheitskritische Netze gewährleisten müssen. Die formale Verifizierung der Datenkommunikation im Hinblick auf die worst-case Deadlines ist grundlegend für die Zertifizierung von neu entwickelten x-by-Wire-Systemen. Diese Verifizierung erlaubt den Start von Flugzeugen, das Lenken von Autos ohne mechanische Verbindung und den Betrieb sicherheitskritischer Industrieanlagen. Daher wurden verschiedene Methoden für die worst-case Modellierung und Analyse von Echtzeitsystemen entwickelt. Eine davon ist der Deterministische Network Calculus (DNC), eine vielseitige Technik, die in verschiedenen Bereichen wie Paketvermittlung, Task Scheduling, System on Chip, softwaredefinierte Netzwerke, Netzwerke in Rechenzentren und Netzwerkvirtualisierung eingesetzt werden kann. DNC ist eine Methode zur Ableitung deterministischer Schranken für zwei der vorrangigsten Leistungsmetriken in Kommunikationssystemen: <ul style="list-style-type: none"> • die Ende-zu-Ende-Verzögerung von Datenflüssen und • der Speicherplatz, den ein Server benötigt, um alle eingehenden Daten in einer Warteschlange zu puffern. 					
Lehrformen Vorlesung mit Übung					
Prüfungsformen Mündliche Modulabschlussprüfung über 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. Informatik

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 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Developer Centered Security (211050)			Kontaktzeit 45 h	Selbststudium 105 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen 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 Entwickeln 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 gegeben. 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 9 CP	Workload 270 h	Semester siehe Prüfungsordnung	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Effiziente Algorithmen (150320 + 150321)			Kontaktzeit 90 h	Selbststudium 180 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: PD Dr. Daniela Kacso Lehrende: PD Dr. Daniela Kacso					
Verwendung des Moduls M.Sc. Angewandte Informatik M.Sc. IT-Sicherheit/ Informationstechnik					
Vorkenntnisse Die Inhalte der Veranstaltung "Datenstrukturen" bzw. "Informatik 2".					
Lernziele (learning outcomes) Nach dem erfolgreichen Abschluss des Moduls: Die Studierenden <ul style="list-style-type: none"> • kennen, wählen aus und nutzen grundlegende Datenstrukturen und Graphenalgorithmen • sind in der Lage Analysetechniken (Korrektheitsbeweise und Laufzeitanalyse) zu erläutern und zu beurteilen • können auch bei praktischen Problemen entscheiden, welche der vermittelten Methoden/Algorithmen/Datenstrukturen anwendbar sind und diese nach Effizienz (insb. Laufzeit der Algorithmen) bewerten • können konkrete Anwendungsprobleme modellieren und bei Bedarf diese Algorithmen weiter entwickeln 					
Inhalt Die Lehrveranstaltung kann sowohl in das Gebiet der praktischen als auch in das Gebiet der theoretischen Informatik eingeordnet werden. Die zentralen Themen sind die Folgenden: <ul style="list-style-type: none"> • Berechnung kürzester Pfade in Digraphen • Berechnung eines maximalen Flusses in einem Transportnetzwerk • Berechnung einer optimalen Lösung bei einem Zuordnungsproblem (auch Matching-Problem genannt) Darüber hinaus beschäftigen wir uns mit Anwendungen dieser grundlegenden Probleme.					
Lehrformen Vortrag der Lehrenden in der Vorlesung, Gruppenarbeit in den Übungen					
Prüfungsformen Schriftliche Modulabschlussprüfung (120 Minuten)					
Voraussetzungen für die Vergabe von Credits Bestandene schriftliche Modulabschlussprüfung					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 9/105: M.Sc. Angewandte Informatik					

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

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

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

Titel des Moduls: Foundations of Programming Languages, Verification, and Security					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Foundations of Programming Languages, Verification, and Security			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen 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 (https://mpi-sp-pap-2023.github.io/book), including familiarity with logic, mechanized proofs, and functional programming in the Coq proof assistant.		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Dr. Roberto Blanco Dr. Catalin Hritcu Lehrende: Dr. Roberto Blanco Dr. Catalin Hritcu					
Verwendung des Moduls o Master Computer Science o Master IT-Sicherheit/Netze und Systeme (Wahl oder Wahlpflicht) o Master IT-Sicherheit/Informationstechnik (Wahl oder Wahlpflicht) o Master Mathematik (Nebenfach Informatik)					
Vorkenntnisse					
Lernziele (learning outcomes) After successful completion of this course, students will be able to I understand how to define in Coq the syntax of simple programming languages: variants of a simple imperative language and the simply-typed lambda calculus; I define the big-step and small-step operational semantics of such simple languages; I formally define type systems for such languages as inductive relations; I work out the metatheory of such languages, by proving results such as type soundness; I understand the semantic foundations of Hoare Logic and Relational Hoare Logic; I use Hoare Logic for verifying the correctness of simple imperative programs, both formally in Coq and informally on paper; I understand the semantic foundations of Secure Information Flow Control and Noninterference. I 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)

Titel des Moduls: Fundamentals of Data Science Fundamentals of Data Science					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Fundamentals of Data Science (141213)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr.-Ing. Aydin Sezgin Lehrende: Prof. Dr.-Ing. Aydin Sezgin					
Verwendung des Moduls M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. Informatik					
Vorkenntnisse Mathematik I-IV Systemtheorie I-III Optimierung 					
Lernziele (learning outcomes)					
Inhalt Die Modulnote setzt sich aus zwei Anteilen zusammen: 1.​Note der mündlichen 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 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Fundamentals of GPU Programming (141374)			Kontaktzeit 45 h	Selbststudium 105 h	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Dr. Denis Eremin Lehrende: Dr. Denis Eremin					
Verwendung des Moduls M.Sc. Angewandte Informatik M.Sc. IT-Sicherheit/ Informationstechnik					
Vorkenntnisse C (Pro­gram­mier­spra­che)					
Lernziele (learning outcomes) Die Studierenden erlernen das Programmieren auf Grafikprozessoren (GPUs)					
Inhalt Zu einem bestimmten Zeitpunkt um 2003 stieg die Rechenleistung nicht auf Kosten der Taktfrequenz des Prozessors, sondern durch Erhöhung der Anzahl der auf dem Prozessorchip zugewiesenen Rechenkern. Grafikprozessoren (GPUs) sind die Meister dieser Computer-Hardware-Entwicklung und bieten bis zu Zehntausende einzelner Kerneinheiten. Gleichzeitig wird das GPU-Speichersystem nicht so sehr durch die Kompatibilitätsanforderungen mit älteren Generationen eingeschränkt wie CPU-Speichersysteme. Deswegen zeigen GPUs im Vergleich zu ihren älteren "Bruder" -Zentraleinheiten (CPUs) eine deutlich bessere Rohleistung der Recheneinheiten und des Speichersystems. Ursprünglich für Videobearbeitungsaufgaben entwickelt, wird die enorme Rechenleistung moderner GPUs üblicherweise zur Unterstützung von CPUs oder zur Lösung einer Vielzahl von Rechenproblemen mit (massiv) parallelisierbaren Teilen verwendet, wodurch Teraflops-hohe Rechenleistung kann schon auf Laptop- / Desktop-Computers erzielt werden. Der vorliegende Kurs zeigt, wie CUDA C (Erweiterung der C-Sprache für die GPU-Programmierung) und das entsprechende (sehr flexible!) CUDA-Laufzeit-API-Framework verwendet werden kann, um die Ausführung einiger typischer Programmiermuster um einen Faktor von 10 oder mehr zu beschleunigen das der CPU. Ausgehend vom CUDA-Programmiermodell geht man zum CUDA-Ausführungsmodell über und betrachtet grundlegende konzeptionelle, Software- und Hardwareprobleme, die zum Verständnis der Funktionsweise von GPUs beitragen. Fallstudien zu mehreren Problemen mit massiv parallelen Algorithmen, die in GPUs implementiert sind, werden ebenfalls weiter ausgeführt. Das theoretische Wissen, das in den Vorlesungen vermittelt wird, wird durch eine Vielzahl von praktischen Beispielen untermauert, an denen die SchülerInnen zu Hause arbeiten können.					
Lehrformen Vorlesung mit Übung					
Prüfungsformen 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 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Human Aspects of Cryptography Adoption			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße 30 Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen Keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Martina Angela Sasse Lehrende: Prof. Dr. Martina Angela Sasse					
Verwendung des Moduls Master IT-Sicherheit/ Informationstechnik Master IT-Sicherheit/ Netze und Systeme					
Vorkenntnisse Keine					
Lernziele (learning outcomes) The aim of the lecture is to examine the reasons why <ol style="list-style-type: none"> 1. cryptographic solutions – which experts agree offer good protection against most of the common attacks today – are not adopted by most individuals and organisations, and 2. end-users, developers and system administrators who do use cryptographic solutions in some form frequently make mistakes that undermine the security protection. 					
Inhalt In 1999, Whitten & Tygar's seminal USENIX paper "Why Johnny Can't Encrypt" established that people cannot use PGP encryption correctly, even with a graphical user interface and instruction. Over the past 20 years, there has been a string of Johnny papers on studies trying to encourage adoption or correct usage. The aim of this CASA lecture is to systematically examine the results of these studies and identify effective ways of promoting adoption and enable correct use of cryptography. <ul style="list-style-type: none"> • Usability, utility and technology adoption • Security threat models and people's mental models • Complexity or simplicity – who needs to know what? • Designing frictionless user journeys • Methods for testing and tweaking 					
Lehrformen Vorlesung mit Übung					
Prüfungsformen Mündliche Prüfung					
Voraussetzungen für die Vergabe von Credits Bestandene Modulabschlussprüfung					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 5/91: M.Sc. IT-Sicherheit/ Informationstechnik 5/99: M.Sc. IT-Sicherheit/ Netze und Systeme					

Titel des Moduls: Information Theory Information Theory					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Information Theory (211007)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Michael Walter Lehrende: Prof. Dr. Michael Walter					
Verwendung des Moduls B.Sc. Informatik B.Sc. IT-Sicherheit M.Sc. Informatik M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Netze und Systeme M.Sc. Angewandte Informatik					
Vorkenntnisse Familiarity with discrete probability (we will briefly remind you of the most important facts). Some experience with precise mathematical statements and rigorous proofs (since we'll see many of those in the course). Part of the homework will require programming in Python.					
Lernziele (learning outcomes) You will learn fundamental concepts, algorithms, and results in information theory. After successful completion of this course, you will know the mathematical model of information theory, how to design and analyze algorithms for a variety of information processing tasks, and how to implement them in Python. You will have independently read about a topic in information theory and presented it to your peers. You will be prepared for an advanced course or a research or thesis project in this area. Please see the course homepage for a precise list of learning objectives.					
Inhalt This course will give an introduction to information theory – the mathematical theory of information. Ever since its inception, information theory has had a profound impact on society. It underpins important technological developments, from reliable memories to mobile phone standards, and its versatile mathematical toolbox has found use in computer science, machine learning, physics, electrical engineering, mathematics, and many other disciplines. Starting from probability theory, we will discuss how to mathematically model information sources and communication channels, how to optimally compress information, and how to design error-correcting codes that allow us to reliably communicate over noisy communication channels. We will also see how techniques used in information theory can be applied more generally to make predictions from noisy data.					
Tentative syllabus:					
<ul style="list-style-type: none"> • Welcome, Introduction to Information Theory • Probability Theory Refresher • Numerical Random Variables, Convexity and Concavity, Entropy • Symbol Codes: Lossless Compression, Huffman Algorithm • Block Codes: Shannon's Source Coding Theorem, its Proof, and Variations 					

- Stream Codes: Lempel-Ziv Algorithm
- Stream Codes: Arithmetic Coding
- Joint Entropies & Communication over Noisy Channels
- Shannon's Noisy Coding Theorem
- Proof of the Noisy Coding Theorem
- Proof of the Converse, Shannon's Theory vs Practice
- Reed-Solomon Codes
- Message Passing for Decoding and Inference, Outlook
- Student Presentations

Please see the course homepage https://qi.rub.de/it_ss23 for more information.

Lehrformen

Lecture with Exercise

Prüfungsformen

Wird noch nachgeliefert

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 Complexity Theory					
Modul-Nr./Code	Credits 9 CP	Workload 270 h	Semester siehe Prüfungsordnung	Turnus Unregelmäßig (i.d.R Sommersemester)	Dauer 1 Semester
Lehrveranstaltungen Komplexitätstheorie			Kontaktzeit 90 h	Selbststudium 180 h	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Thomas Zeume Lehrende: Prof. Thomas Zeume					
Verwendung des Moduls M.Sc. Informatik M.Sc. Angewandte Informatik					
Vorkenntnisse <p>Kenntnisse aus einem Grundkurs in theoretischer Informatik (Grundlagen der Komplexitätstheorie einschließlic NP-Vollständigkeit und Reduktionen) werden erwartet.</p>					
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 lösbar 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: <ul style="list-style-type: none"> • 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 Abschlussprüfung; mündlich, 20-30min					
Voraussetzungen für die Vergabe von Credits Bestandene mündliche Modulabschlussprüfung					

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

9/105: M.Sc. Angewandte Informatik [PO22]

9/97: M.Sc. Informatik [PO 23]

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

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

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

5/105: M.Sc. Angewandte Informatik

Titel des Moduls: Künstliche Neuronale Netze Artificial Neural Networks					
Modul-Nr./Code	Credits 6 CP	Workload 180 h	Semester siehe Prüfungsordnung	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Artificial Neural Networks (212006)			Kontaktzeit 60 h	Selbststudium 120 h	Gruppengröße 150 Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Sen Cheng Lehrende: Prof. Dr. Sen Cheng					
Verwendung des Moduls B.Sc. Angewandte Informatik B.Sc. Informatik M.Sc. IT-Sicherheit/ Informationstechnik					
Vorkenntnisse Grundkenntnisse in der Infinitesimalrechnung, linearen Algebra, Statistik und Informatik. Erfahrung mit einer Programmiersprache.					
Lernziele (learning outcomes) Die mathematischen Grundlagen, Möglichkeiten und Beschränkungen überwachter Lernverfahren für Regression und Klassifikation mit künstlichen neuronalen Netzen (KNN), sowie für deren Anwendung erforderliche praktische Kenntnisse werden vermittelt. Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • verstehen Studierende die theoretisch-mathematischen Grundlagen von KNN im Kontext des überwachten Lernens. • können Studierende selbstständig zwischen verschiedenen KNN unterscheiden und in einer Anwendungssituation das geeignete Verfahren auswählen. • können Studierende grundlegende Verfahren selbstständig in einer höheren Programmiersprache implementieren, sowie ihre eigene Implementierung und Standard- Implementierung anderer auf Daten anwenden. • können Studierende Ergebnis der KNN selbstständig interpretieren, insbesondere erkennen, wann sie unrealistisch sind. 					
Inhalt Verfahren: Struktur von Optimierungsproblemen, Regression, logistische Regression, biologische neuronale Netze, Modellselektion, universelle Approximationstheorem, Perzeptron, mehr-schichtiges Perzeptron, Backpropagation, tiefe neuronale Netze, rekurrente neuronale Netze, Long-Short Term Memory, Hopfield Netze, Boltzmann-Machine Software: python, numpy, matplotlib, scikit-learn, tensorflow					
Lehrformen Vorlesung, Hausaufgaben, angeleitete Übungen am Computer					
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)					

6/168: B.Sc. Angewandte Informatik [PO 22]

6/170: B.Sc. Angewandte Informatik [PO 20]

6/158: B.Sc. Informatik [PO 22]

6/170: B.Sc. Informatik [PO 20]

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

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

Titel des Moduls: Machine Learning: Supervised Methods Machine Learning: Supervised Methods					
Modul-Nr./Code	Credits 6 CP	Workload 180 h	Semester siehe Prüfungsordnung	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Machine Learning: Supervised Methods (211024)			Kontaktzeit 60 h	Selbststudium 120 h	Gruppengröße 80 Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Tobias Glasmachers Lehrende: Prof. Dr. Tobias Glasmachers					
Verwendung des Moduls Master Informatik Master Angewandte Informatik Master IT-Sicherheit/ Informationstechnik					
Vorkenntnisse <p>empfohlen: Vorlesung "Mathematics for Modeling and Data Analysis"</p>					
Lernziele (learning outcomes) Internationalisierung: Die Veranstaltung wird auf Englisch durchgeführt. Digitalisierung: Inhalte werden durch Videos und Lesematerial vermittelt. Übungsaufgaben mit Programmieranteilen werden in Form von Jupyter-Notebooks bereitgestellt. Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • verstehen die Teilnehmer die Grundlagen der statistischen Lerntheorie • kennen die Teilnehmer die wichtigsten Algorithmen des überwachten statistischen Lernens und können diese auf Lernprobleme anwenden, • kennen die Teilnehmer Stärken und Beschränkungen verschiedenen Lernverfahren, • können die Teilnehmer Standardsoftware zum maschinellen Lernen zur Lösung neuer Probleme einsetzen. 					
Inhalt Grundlagen der statistischen Lerntheorie, Querschnitt der wichtigsten Algorithmen des maschinellen Lernens, konkrete Problemlösung mit Standardsoftware					
Lehrformen flipped classroom					
Prüfungsformen Abschlussprüfung; Klausur 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. Informatik 6/91: M.Sc IT-Sicherheit/ Informationstechnik					

Titel des Moduls: Master Praktikum/Projektarbeit IT-Sicherheit

Modul-Nr./Code	Credits 4 CP	Workload 120 h	Semester 3	Turnus jedes Semester	Dauer 1 Semester
Lehrveranstaltungen <ul style="list-style-type: none"> • Praktische Kryptanalyse von symmetrischen Chiffren (211401) • Projekt Netz- und Datensicherheit (212412) • Forschungspraktikum Human-Centred Security (212408) • Initial Research in Information Security (212402) • Praktikum TLS Implementierung (212414) • Praktikum zur Hackertechnik (Hackerpraktikum) (212413) • Research in Software/Internet Security (2124) • Master-Praktikum ARM Processors for Embedded Cryptography (212407) • Developer Centered Security (212417) • Praktikum Wireless Physical Layer Security (142025) 			Kontaktzeit je nach Veranstaltungswahl	Selbststudium abhängig von der Praktikumswahl	Gruppengröße Studierende
Unterrichtssprache abhängig von der Praktikumswahl: Deutsch oder Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Studiendekan IT-Sicherheit Lehrende: siehe Praktikumsbeschreibung					
Verwendung des Moduls M.Sc. IT-Sicherheit / Informationstechnik M.Sc. IT-Sicherheit / Netze und Systeme					
Vorkenntnisse abh#228;ngig vom gew#228;hlten Praktikum					
Lernziele (learning outcomes) Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • 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 aktuell Praktika zu folgenden Themen angeboten: <ul style="list-style-type: none"> • 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.

Lehrformen

Praktikum im Block oder als semesterbegleitende Veranstaltung

Prüfungsformen

Praktikum

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

unbenotet

Titel des Moduls: Menschliches Verhalten in der IT-Sicherheit					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Menschliches Verhalten in der IT-Sicherheit (211033)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen Keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Martina Angela Sasse Lehrende: Prof. Dr. Martina Angela Sasse M. Sc. Jonas Hielscher					
Verwendung des Moduls M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Netze und Systeme					
Vorkenntnisse Der vorherige Besuch der Vorlesung "Einführung in die Usable Security and Privacy" wird empfohlen					
Lernziele (learning outcomes) Die Veranstaltung vermittelt theoretische und praktische Kenntnisse über Forschungs- methoden im Bereich usable Security mit einem besonderen Schwerpunkt auf Laborstudien. Es werden theoretische Kenntnisse vermittelt, auf deren Grundlage die Studierenden selbstständig eine Laborstudie planen und umsetzen und auf diese Weise praktische Kenntnisse erwerben sollen.					
Inhalt In <i>Menschliches Verhalten in der IT-Sicherheit</i> lernt ihr, welche Faktoren Einfluss auf das Sicherheitsverhalten von Angestellten in Unternehmen und Nutzenden im Alltag nehmen, und welche Möglichkeiten bestehen, dieses zu beeinflussen und verändern. Außerdem wird vermittelt, warum bestehende Ansätze des Information Security Management (auch nach ISO 27000) in der Praxis oft nicht funktionieren und wie wir sie erweitern bzw. anpassen sollten. Studierende werden befähigt IT-Sicherheit in Organisationen aus einem ganzheitlichen Ansatz heraus zu betrachten, was unter anderem zwingend erforderlich ist um später Sicherheitsführungsaufgaben wahrzunehmen. Die Vorlesungsinhalte sind dabei umfangreich mit Erfahrungen aus der Praxis angereichert.					
Lehrformen Vorlesung und Übung					
Prüfungsformen Schriftliche Modulabschlussprüfung (120 Minuten)					
Voraussetzungen für die Vergabe von Credits Bestandene Modulabschlussprüfung					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 5/99: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 22] 5/96: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 20] 5/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22] 5/84: M.Sc. IT-Sicherheit/ Informationstechnik [PO 20]					

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

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Message-Level Security (212060)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		

Modulbeauftragte/r und hauptamtlich Lehrende

Modulbeauftragte/r: Dr.-Ing. Christan Mainka

Lehrende: Dr.-Ing. Christan Mainka

Dr.-Ing. Vladislav Mladenov

Verwendung des Moduls

B.Sc. IT-Sicherheit/ Informationstechnik

M.Sc. IT-Sicherheit/ Informationstechnik

M.Sc. IT-Sicherheit/ Netze und Systeme

Vorkenntnisse

Grundkenntnisse: HTTP, HTML und Kryptografie

Grundkenntnisse: der englischen Sprache, da dies die Sprache von Folien, Sätzen, Bildern und der Virtuellen

Medien sind

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-Szenarien 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					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen 212064 - Microarchitectural Attacks and Defenses			Kontaktzeit 90 h	Selbststudium	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen none		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Yuval Yarom Lehrende: Prof. Yuval Yarom					
Verwendung des Moduls Master ITS - Informationstechnik Master ITS - Netze und Systeme					
Vorkenntnisse The course assumes that students can program in C or learn the language as they go. They need enough experience to be able to program on remote machines, using SSH. Basic understanding of how computers work, assembly language, and the role of the operating system is required. Understanding of basic concepts in computer security (security domains, vulnerabilities, etc.) and familiarity with basic cryptography (AES, RSA, ECC) is helpful.					
Lernziele (learning outcomes) • Diagnose microarchitectural vulnerabilities • Assess software for resilience against microarchitectural vulnerabilities • Design and program proof-of-concept exploits of vulnerable software and hardware • Design and implement countermeasures for software executing on vulnerable hardware					
Inhalt The course covers the area of microarchitectural attacks and defences. It starts with cache attacks, covering the main techniques (Prime+Probe, Evict+Time, and Flush+Reload). Building on this basis it explores variants of the attacks targetting other storage elements as well as attacks that exploits bandwidth limitations. In parallel with exploring these attacks, the course will describe various countermeasures, with special focus on constant-time programming. The course then switches to speculative execution attacks, identifying and classifying the various attacks, defences, and counter-attacks. The course further covers several related attacks, including Rowhammer and voltage- and frequency-based attacks. Additionally, the course pays special attention to attack scenarios exploring, in particular, attacks on the operating system kernel, web-based and other remote attacks, and attacks on trusted execution environments. The course puts special focus on practical implementation of both attack and defence techniques.					
Lehrformen Lecture with excercise					
Prüfungsformen Assignments and practical projects.					
Voraussetzungen für die Vergabe von Credits					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)					

Titel des Moduls: Nebenläufige Programmierung

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Nebenläufige Programmierung (211012)			Kontaktzeit 45 h	Selbststudium 105 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen 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 <ul style="list-style-type: none">• 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 <ul style="list-style-type: none">• Anwendungen vs. Prozesse• Programme und ihre Ausführung• Vorteile und Probleme von nebenläufigen Programmen (Verbesserung der Performanz, 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: Processor Security Processor Security					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Sommersemester	Dauer Semester
Lehrveranstaltungen Processor Security (211099)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Dr.-Ing. Pascal Sasdrich Lehrende: Dr.-Ing. Pascal Sasdrich					
Verwendung des Moduls B.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Netze und Systeme					
Vorkenntnisse Inhalte der Module „Informatik 1 – Programmierung“ und „Technische Informatik 1 – Rechnerarchitektur“					
Lernziele (learning outcomes) Im Rahmen dieser Veranstaltung lernen die Studierenden wichtige Sicherheitsaspekte und -konzepte moderner Prozessoren kennen. Der Fokus der Veranstaltung liegt dabei auf (a) Kenntnis gängiger Angriffsvektoren, (b) Verständnis der zugrundeliegenden Hardware- und Prozessormechanismen, (c) Diskussion möglicher Gegenmaßnahmen, sowohl in Hardware als auch Software.					
Inhalt Moderne Prozessorenarchitekturen, von eingebetteten Mikrocontrollern bis hin zu Server-CPU's, bilden das Kernstück unserer heutigen Informationsgesellschaft und werden seit Jahrzehnten immer komplizierter. Diese gesteigerte Komplexität führt aber unausweichlich zu neuen Schwachstellen und gesteigerter Anfälligkeiten gegen gezielte Angriffe. Im Rahmen dieser Veranstaltung werden daher verschiedene Sicherheitsaspekte und -konzepte moderner Prozessorarchitekturen vorgestellt und erläutert. Dazu werden sowohl wichtige Angriffsvektoren (z.B. Buffer Overflows, Privilege Escalation, Control-Flow Manipulation, Side Channel Attacks, Microarchitectural Attacks, ...), fundamentale Ursachen in der Prozessorarchitektur, als auch mögliche Abwehrstrategien diskutiert. Vorlesungsbegleitend wird ein Moodle-Kurs angeboten, der zusätzliche Inhalte sowie die praktischen Übungen bereithält.					
Lehrformen Vorlesung mit Übung					
Prüfungsformen Schriftliche Modulabschlussprüfung (120 Minuten)					
Voraussetzungen für die Vergabe von Credits					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 5/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22] 5/84: M.Sc. IT-Sicherheit/ Informationstechnik [PO 20]					

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

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

Titel des Moduls: Programmanalyse					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Programmanalyse (211015)			Kontaktzeit 60 h	Selbststudium	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Kevin Borgolte Lehrende: Prof. Kevin Borgolte					
Verwendung des Moduls M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Netze und Systeme					
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: <ul style="list-style-type: none"> • Statische und dynamische Analyse von Programmen • Analyse von Kontroll- und Datenfluss • Symbolische Ausführung • Taint Tracking • Binary Instrumentation • Program Slicing • Überblick zu existierenden Analysetools Daneben wird im ersten Teil der Vorlesung eine Einführung in x86/x64 Assembler gegeben sowie die grundlegenden Techniken aus dem Themenbereich Reverse Engineering vorgestellt. Begleitet wird die Vorlesung von Übungen, in denen die vorgestellten Konzepte und Techniken praktisch eingeübt werden.					
Lehrformen Vorlesung mit Übung					
Prüfungsformen mündliche oder schriftliche Modulabschlussprüfung (wird zu Beginn des 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 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Proofs are Programms (211003)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße 40 Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Dr. Catalin Hritcu Lehrende: Dr. Catalin Hritcu Dr. Clara Schneidewind					
Verwendung des Moduls 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 <ul style="list-style-type: none"> • 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 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Sommersemester	Dauer Semester
Lehrveranstaltungen Public Key Kryptanalyse 1 (211055)			Kontaktzeit 45 h	Selbststudium 105 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen 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 & Informatiker) und ein Interesse an algorithmischen Techniken und Kryptographie, in Theorie und Praxis (umgesetzt mit Hilfe des Computeralgebra-Systems Sage).					
Lernziele (learning outcomes) Die Studierenden sollen breite Kenntnisse zu algorithmischen Techniken der asymmetrischen Kryptanalyse, insbesondere für codierungsbasierte Kryptographie, erlangen. Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • kennen die Studierenden grundlegende Schlüsselfindungs-Algorithmen wie Brute-Force und Meet-in-the-Middle und können diese auf neue kryptographische Systeme anwenden, • beherrschen sie die Grundlagen linearer Codes und ihrer Dualcodes, insbesondere als kryptographische Anwendung das McEliece-Kryptosystem, • kennen Studierende Time-Memory Techniken wie Pollard Rho und Parallel Collision Search, und können sie auf neue Probleme anwenden, • haben Studierende einen Überblick über alle aktuellen Dekodieralgorithmen im Bereich des Information Set Decoding, die für die Sicherheits-Evaluierung moderner kodierungsbasierter Kryptosysteme relevant sind, • sind Studierende in der Lage, Techniken der Kryptanalyse mit Hilfe der Computer-Algebra Sage zu implementieren. 					
Inhalt Kryptanalyse dient dazu, kryptographische Systeme derart zu instantiiieren, dass sie einerseits ein vordefiniertes Sicherheitsniveau bieten, andererseits aber möglichst performant sind. Die Kryptanalyse bietet dazu einen ganzen Werkzeugkoffer an algorithmischen Techniken, um die Evaluation neuer kryptographischer Systeme zu realisieren. Dies beinhaltet sowohl klassische Algorithmen als auch Algorithmen für Quantenrechner, damit die verwendete Kryptographie selbst in einer Ära von Quantenrechnern sicher bleiben.					
Lehrformen Die Vorlesung wird als seminaristischer Unterricht abgehalten, die praktischen Übungen am Rechner mit der Computer-Algebra Sage werden zudem weitere Lehrformen wie Gruppen- und Projektarbeit beinhalten.					
Prüfungsformen Schriftliche Modulabschlussprüfung über 120 Minuten					
Voraussetzungen für die Vergabe von Credits					
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					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Public Key Verschlüsselung			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Jun. Prof. Dr. Nils Fleischhacker Lehrende: Jun. Prof. Dr. Nils Fleischhacker					
Verwendung des Moduls M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Netze und Systeme					
Vorkenntnisse Als Voraussetzung für die Vorlesung sind Vorkenntnisse in Kryptographie ¨ und beweisbarer Sicherheit, insbesondere von Reduktionsbeweisen, hilfreich aber nicht zwingend erforderlich.					
Lernziele (learning outcomes) Die Studierenden haben einen Einblick in in theoretische und praktische Aspekte der Public Key Verschlüsselung erhalten					
Inhalt Die Vorlesung gibt einen Einblick in theoretische und praktische Aspekte der Public Key Verschlüsselung. Dies umfasst Grundlagen und formalen Definitionen von Sicherheit (CPA, CCA1, CCA2), die beweisbare Sicherheit verschiedener theoretischer und praktischer Konstruktionen, sowie die Verbindungen von Public Key Verschlüsselung zu anderen Aspekten der Kryptographie.					
Lehrformen Vorlesung mit Übung					
Prüfungsformen Mündlich (30 Minuten)					
Voraussetzungen für die Vergabe von Credits Bestandene Modulabschlussprüfung					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 5/91: M.Sc. IT-Sicherheit/ Informationstechnik 5/99: M.Sc. IT-Sicherheit/ Netze und Systeme					

Titel des Moduls: Quantum Cryptography Quantum Cryptography					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Quantum Cryptography (212016)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Michael Walter Lehrende: Prof. Michael Walter Dr. Giulio Malavolta					
Verwendung des Moduls M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Netze und Systeme					
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

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

Titel des Moduls: Software Protection
Software Protection

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Software Protection (211107)			Kontaktzeit 45 h	Selbststudium 105 h	Gruppengröße Studierende

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

Modulbeauftragte/r und hauptamtlich Lehrende
 Modulbeauftragte/r: Professur für Systemsicherheit
 Lehrende: Dr.-Ing. Tim Blazytko
 Philipp Koppe

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

Vorkenntnisse
 Im Bereich Reverse Engineering sind empfohlen, beispielsweise durch Erfahrung mit x86-Assembler. Erfahrung in systemnaher Programmierung (Assembler, C) ist hilfreich.

Lernziele (learning outcomes)
 Die Studierenden kennen verschiedene Konzepte, Techniken und Tools aus dem Bereich Software Protection. Dies beinhaltet sowohl Wissen über das Design und die Implementierung von Obfuskerungstechniken als auch die Sicherheitsanalyse gängiger Systeme. Die Studierenden lernen erweiterte Techniken zur Programmanalyse, mit welchen sie komplexe Protection-Mechanismen angreifen können. Sie sind in der Lage, verschiedene Aspekte der Software Protection zu beschreiben und auf neue Problemstellungen anzuwenden.

Inhalt
 Unter Software Protection versteht man Maßnahmen, welche die Analyse bzw. das Reverse Engineering von Software erschweren. Solche Methoden finden sowohl Anwendung in kommerzieller Software, um Piraterie zu verhindern, als auch in Malware, um deren Funktionsweise zu verschleiern.

In dieser Lehrveranstaltung lernen die Studierenden gängige Methoden der Software Protection kennen sowie Methoden, um diese zu brechen. Dazu designen und implementieren sie in praxisnahen Aufgaben erst ihre eigenen Protection-Mechanismen, welche sie im Anschluss brechen werden mit dem Ziel, diese wieder zu verbessern. Parallel dazu werden Schutzmechanismen aus der echten Welt analysiert, attackiert und diskutiert.

Dabei werden unter anderem die folgenden Themen und Techniken aus dem Bereich Software Protection behandelt:

- Opaque Predicates
- Control-flow Flattening
- Mixed Boolean-Arithmetic Expressions
- Virtual Machines
- Anti-Tamper
- Symbolische Ausführung

- SMT Solving
- Programmsynthese
- Überblick zu existierenden Analysetools und Frameworks

Lehrformen

Vorlesung mit Übung

Prüfungsformen

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

Voraussetzungen für die Vergabe von Credits

Erfolgreiche Kompetenznachweis im Semester

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

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

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

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

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

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

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

Titel des Moduls: Software Security Software Security					
Modul-Nr./Code	Credits 9 CP	Workload 270 h	Semester siehe Prüfungsordnung	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Software Security (212026)			Kontaktzeit 60 h	Selbststudium 210 h	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen Keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Kevin Borgolte Lehrende: Prof. Kevin Borgolte					
Verwendung des Moduls M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Netze und Systeme M.Sc. Angewandte Informatik					
Vorkenntnisse Keine					
Lernziele (learning outcomes)					
Inhalt					
Lehrformen Vorlesung mit Übung					
Prüfungsformen Schriftliche Hausarbeit (Take-Home-Exam) am Ende der Vorlesungszeit und ggf. mündlicher Vortrag.					
Voraussetzungen für die Vergabe von Credits Bestandene Hausarbeit und ggf. mündliche Vortrag.					
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 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Software-Implementierung kryptographischer Verfahren (211035)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr.-Ing. Tim Guneyusu Lehrende: Dr.-Ing. Max Hoffmann					
Verwendung des Moduls 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: <ul style="list-style-type: none"> • Effiziente Implementierung von Blockchiffren • Bitslicing • Effiziente Arithmetik in $GF(2^m)$ • Effiziente Arithmetik auf elliptischen Kurven • Spezielle Primzahlen zur schnellen modularen Reduktion • Primzahltests • Post-Quantum Kryptographie • Secure Coding 					
Lehrformen Vorlesung mit Übungen					
Prüfungsformen Schriftliche Modulabschlussprüfung (120 Minuten)					
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 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Symmetrische Kryptanalyse			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Nils-Gregor Leander Lehrende: Prof. Dr. Nils-Gregor Leander					
Verwendung des Moduls M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Netze und Systeme M.Sc. Angewandte Informatik					
Vorkenntnisse <p>Inhalt der Vorlesung "Einführung in die Kryptographie 1"</p>					
Lernziele (learning outcomes) Die Studierenden haben ein vertieftes Verständnis für die Sicherheit symmetrischer Chiffren.					
Inhalt Wir behandeln die wichtigsten Themen in der symmetrischen Kryptanalyse. Nach einer ausführlichen Vorstellung von linearer und differentieller Kryptanalyse werden weitere Angriffe auf symmetrische Primitive, insbesondere Block-Chiffren behandelt. Hierzu zählen insbesondere Integral (auch Square) Attacks, Impossible Differentials, Boomerang-Angriffe und Slide-Attacks. Neben den Angriffen selbst werden auch immer die daraus resultierenden Design-Kriterien beschrieben, um neue Algorithmen sicher gegen die Angriffe zu machen.					
Lehrformen					
Prüfungsformen Mündliche Modulabschlussprüfung (30 Minuten)					
Voraussetzungen für die Vergabe von Credits Bestandene mündliche Modulabschlussprüfung.					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 5/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22] 5/84: M.Sc. IT-Sicherheit/ Informationstechnik [PO 20] 5/99: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 22] 5/96: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 20] 5/105: M.Sc. Angewandte Informatik					

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

Modul-Nr./Code	Credits 3 CP	Workload 90 h	Semester	Turnus jedes Semester	Dauer Semester
Lehrveranstaltungen			Kontaktzeit	Selbststudium	Gruppengröße Studierende
211104 Human Centred Security and Privacy			30 h		
211110 Seminar Real-World Kryptanalyse					
211114 Master-Seminar on Security and Privacy of Mobile Operating Systems					
211117 Seminar Satisfiability (bis SoSe 23)					
211119 Quantum Algorithms (bis SoSe 23)					
211121 Fortgeschrittene Themen des Model Checking ()					
211122 Seminar über Grenzen in der theoretischen Informatik ()					
211129 Master-Seminar Developer Centered Security					
211132 Master-Seminar Digitale Souveränität					
211133 Seminar on Current Topics for Systems Security and Privacy					
212109 Information Security Seminar					
212111 Seminar Ressourceneffiziente Systemsoftware					
212112 Seminar Security Engineering					
212118 Seminar zur symmetrischen Kryptographie					
212121 Seminar Netz- und Datensicherheit					
212122 Seminar Current Topics in Device Firmware Security					
212125 Software and Internet Security Seminar					
212126 Seminar Implementation Security (bis SoSe 23)					
Unterrichtssprache Deutsch oder Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Studiendekan IT-Sicherheit Lehrende: siehe jeweiliges Seminar					
Verwendung des Moduls B.Sc. IT-Sicherheit					
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)**

3/149: B.Sc. IT-Sicherheit [PO 20]

3/150: B.Sc. IT-Sicherheit [PO 22]

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

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

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