UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

AMTLICHE MITTEILUNGEN

VERKÜNDUNGSBLATT DER UNIVERSITÄT PADERBORN AM.UNI.PB

AUSGABE 68.21 VOM 23. DEZEMBER 2021

ZWEITE SATZUNG ZUR ÄNDERUNG DER PRÜFUNGSORDNUNG FÜR DEN MASTERSTUDIENGANG INFORMATIK DER FAKULTÄT FÜR ELEKTROTECHNIK, INFORMATIK UND MATHEMATIK AN DER UNIVERSITÄT PADERBORN

VOM 23. DEZEMBER 2021

HERAUSGEBER: PRÄSIDIUM DER UNIVERSITÄT PADERBORN

Zweite Satzung zur Änderung der Prüfungsordnung für den Masterstudiengang Informatik der Fakultät für Elektrotechnik, Informatik und Mathematik an der Universität Paderborn

vom 23. Dezember 2021

Aufgrund des § 2 Abs. 4 und des § 64 Abs. 1 des Gesetzes über die Hochschulen des Landes Nordrhein-Westfalen (Hochschulgesetz – HG) vom 16. September 2014 (GV.NRW. S. 547), zuletzt geändert durch Artikel 1 des Gesetzes vom 25. November 2021 (GV. NRW. Seite 1210a), hat die Universität Paderborn die folgende Satzung erlassen:

Artikel I

Die Prüfungsordnung für den Masterstudiengang Informatik an der Universität Paderborn in der Fassung vom 16. Juni 2017 (AM.Uni.Pb. 44.17), geändert durch Satzung vom 12. Oktober 2018 (AM.Uni.Pb. 35.18), wird wie folgt geändert:

1. In § 10 Absatz 1 Satz 1 wird nach Nummer 6 „Data Science“ folgende Angabe angefügt: „7. Security”.
2. Anhang 1: Module und Prüfungsformen wird wie folgt geändert:
   a) In der ersten Tabelle wird nach der Übersicht zur Focus Area „Data Science“ folgende Übersicht eingefügt:

Focus Area Modul	**LP** **Modul** SWS	**Prüfungsformen**	**Bemerkung**
Security	***6***	Mündliche Prüfung oder Klausur als Modulabschlussprüfung	Wahlpflichtmodul Im Falle der Einschreibung mit Auflagen mit den Auflagenkursen • **Mathematik 1 und 2 und/oder** • **Modelle und Algorithmen 1 und 2** ist das Bestehen der zugehörigen Prüfungen bis zur Prüfungsanmeldung des Moduls nachzuweisen. Voraussetzung zur Teilnahme an der Modulprüfung: Studienleistung
Eines der Module • Advanced Distributed Algorithms and Data Structures • Designing code analyses for large-scale software systems 1 • Designing code analyses for large-scale software systems 2 • Foundations of Cryptography • Introduction to Quantum Computation • Quantum Complexity Theory • Real World Crypto Engineering • Web Security	3+2		

b) Die zweite Tabelle wird wie folgt gefasst:

Bei einer **Einschreibung mit Auflagen** gemäß § 4 müssen gegebenenfalls Auflagenkurse absolviert werden, bevor die Modulabschlussprüfung eines Moduls abgelegt werden kann. Es bestehen die folgenden Auflagenkurse:

Auflagenkurs	Umfang	Voraussetzung für Module der Focus Area
Grundlagen Mensch-Maschine-Wechselwirkung	2 SWS	aus der Focus Area Software Engineering: Kontextuelle Informatik
Mathematik 1 und 2	4 SWS	Algorithm Design Networks and Communication Intelligence and Data Data Science Security
Modelle und Algorithmen 1 und 2	4 SWS	Algorithm Design Intelligence and Data Data Science Security
Soft Skills, Management	2 SWS	Seminar I und II Projektgruppe
Software Engineering 1 und 2	4 SWS	Software Engineering mit Ausnahme des Moduls Kontextuelle Informatik Intelligence and Data Data Science
Systems 1 und 2	4 SWS	Networks and Communication Computer Systems

3. Anhang 4 wird wie folgt geändert:

a) Im Kapitel 2 Studienrichtung wird folgender Punkt 2.7 eingefügt:

Security

Koordination: Prof. Dr. Eric Bodden

Enthaltene Module:

- Advanced Distributed Algorithms and Data Structures
- Designing code analyses for large-scale software systems 1
- Designing code analyses for large-scale software systems 2
- Foundations of Cryptography
- Introduction to Quantum Computation
- Quantum Complexity Theory
- Real World Crypto Engineering
- Web Security

Beschreibung:

In allen Lebensbereichen bieten digitale Technologien, wie zum Beispiel das (Industrial) Internet of Things, Cyber-Physical Systems, Digital Automotives, Digital Health oder Industrie 4.0, ein immenses Innovationspotenzial. Die zunehmende Digitalisierung erfordert jedoch neue Ansätze, um dieses Potenzial sicher nutzen zu können. Um diese Herausforderung angehen zu können, besteht in Industrie, Forschung und Lehre ein großer Bedarf an gut ausgebildeten Informatik-Experten mit fundierten Kenntnissen in der IT-Sicherheit. Im Vertiefungsgebiet "Security" wird ein solides theoretisches Grundwissen in Kombination mit praktischen Fertigkeiten vermittelt. Das Lehrangebot deckt fachliche Kompetenzen aus dem Bereich der IT-Sicherheit ab (z.B. Softwaresicherheit, formale Verifikation, Grundlagen der modernen Kryptographie und Kommunikationssicherheit) ab, in denen typische Sicherheitslücken und Angriffstechniken vorgestellt werden und Gegenmaßnahmen und ihre Wirksamkeit untersucht werden.

Da Sicherheit nicht unabhängig von konkreten Anwendungen gesehen werden kann und unterschiedliche Anwendungen unterschiedliche Sicherheitsanforderungen haben, werden darüber hinaus auch fachliche Kompetenzen in modernen Anwendungsfeldern mit besonderen Sicherheitsanforderungen (z.B. Kommunikationsprotokolle in den Bereichen Mobile und Automotive) sowie ergänzende Qualifikationen in den Bereichen Algorithmen und Quanten-Computing abgedeckt.

b) In Kapitel 3 Module werden die aus dem Anhang zu dieser Änderungssatzung ersichtlichen Modulbeschreibungen angefügt.

Artikel II

(1) Diese Änderungssatzung tritt am 1. April 2022 in Kraft. Sie wird in den Amtlichen Mitteilungen der Universität Paderborn (AM.Uni.Pb.) veröffentlicht.

(2) Gemäß § 12 Absatz 5 HG kann nach Ablauf eines Jahres seit der Bekanntmachung dieser Ordnung gegen diese Ordnung die Verletzung von Verfahrens- oder Formvorschriften des Hochschulgesetzes oder des Ordnungs- oder des sonstigen autonomen Rechts der Hochschule nicht mehr geltend gemacht werden, es sei denn,

1. die Ordnung ist nicht ordnungsgemäß bekannt gemacht worden,
2. das Präsidium hat den Beschluss des die Ordnung beschließenden Gremiums vorher beanstandet,
3. der Form- oder Verfahrensmangel ist gegenüber der Hochschule vorher gerügt und dabei die verletzte Rechtsvorschrift und die Tatsache bezeichnet worden, die den Mangel ergibt, oder
4. bei der öffentlichen Bekanntmachung der Ordnung ist auf die Rechtsfolge des Rügeausschlusses nicht hingewiesen worden.

Ausgefertigt aufgrund des Beschlusses des Fakultätsrates der Fakultät für Elektrotechnik, Informatik und Mathematik vom 15. November 2021 und der Rechtmäßigkeitsprüfung durch das Präsidium der Universität Paderborn vom 15. Dezember 2021.

Paderborn, den 23. Dezember 2021

Die Präsidentin
der Universität Paderborn

Professorin Dr. Birgitt Riegraf

Advanced Distributed Algorithms and Data Structures			
Advanced Distributed Algorithms and Data Structures			
Modulnummer:	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** Wintersemester
	Studiensemester:	**Dauer (in Sem.):** 1	**Sprache:** en

1 **Modulstruktur**

	Lehrveranstaltung	**Lehrform**	**Kontaktzeit (h)**	**Selbststudium (h)**	**Status (P/WP)**	**Gruppengröße (TN)**
a)	Advanced Distributed Algorithms and Data Structures	V3 Ü2	75	105	P	30

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

Teilnahmevoraussetzungen der Lehrveranstaltung Advanced Distributed Algorithms and Data Structures:
Empfohlene Vorkenntnisse
Algorithmen und Datenstrukturen, verteilte Algorithmen und Datenstrukturen

4 **Inhalte:**

Die Vorlesung stellt fortgeschrittene Methoden für verteilte Algorithmen und Datenstrukturen vor. Themen sind unter anderem Zugriffskontrolle, Synchronisation, Konsensus, Informationsverbreitung, hybride Netze, Scheduling, und Optimierung. Aufbauend auf Lösungen zu diesen Themen werden auch konkrete Anwendungen vorgestellt.

Inhalte der Lehrveranstaltung Advanced Distributed Algorithms and Data Structures:
Die Vorlesung stellt fortgeschrittene Methoden für verteilte Algorithmen und Datenstrukturen vor. Themen sind unter anderem Zugriffskontrolle, Synchronisation, Konsensus, Informationsverbreitung, hybride Netze, Scheduling, und Optimierung. Aufbauend auf Lösungen zu diesen Themen werden auch konkrete Anwendungen vorgestellt.

5 **Lernergebnisse und Kompetenzen:**

Studierende lernen fortgeschrittene Methoden und Verfahren für aktuell sehr relevante verteilte Systeme kennen. Sie können Verfahren an neue Situationen anpassen und deren Komplexität bestimmen. Sie können grundlegende Verfahren implementieren.
Nichtkognitive Kompetenzen

- Gruppenarbeit
- Lernkompetenz
- Schreib- und Lesekompetenz (wissenschaftlich)
- Selbststeuerungskompetenz

<table>
<tr><td>6</td><td>Prüfungsleistung:
☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)<table><tr><th>zu</th><th>Prüfungsform</th><th>Dauer bzw. Umfang</th><th>Gewichtung für die Modulnote</th></tr><tr><td>a)</td><td>Klausur oder mündliche Prüfung</td><td>90-120 min bzw. 40 min</td><td>100%</td></tr></table>Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Prüfungsleistung konkret zu erbringen ist.</td></tr>
<tr><td>7</td><td>Studienleistung, qualifizierte Teilnahme:<table><tr><th>zu</th><th>Form</th><th>Dauer bzw. Umfang</th><th>SL / QT</th></tr><tr><td>a)</td><td>Schriftliche Übungsaufgaben</td><td></td><td>SL</td></tr></table>Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.</td></tr>
<tr><td>8</td><td>Voraussetzungen für die Teilnahme an Prüfungen:
Bestehen der Studienleistung</td></tr>
<tr><td>9</td><td>Voraussetzungen für die Vergabe von Credits:
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</td></tr>
<tr><td>10</td><td>Gewichtung für Gesamtnote:
Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Masterstudiengang Informatik v3, Masterstudiengang Informatik v4</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Prof. Dr. Christian Scheideler</td></tr>
<tr><td>13</td><td>Sonstige Hinweise:
Hinweise der Lehrveranstaltung Advanced Distributed Algorithms and Data Structures:
Methodische Umsetzung
Vorlesung mit Übungen und Softwareprojekt
Lernmaterialien, Literaturangaben
Skript</td></tr>
</table>

Erzeugt am 26. Oktober 2021 um 14:34.

Designing code analyses for large-scale software systems 1			
Designing code analyses for large-scale software systems 1			
Modulnummer:	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** Wintersemester
	Studiensemester:	**Dauer (in Sem.):** 1	**Sprache:** en

1 **Modulstruktur**

	Lehrveranstaltung	**Lehrform**	**Kontaktzeit (h)**	**Selbststudium (h)**	**Status (P/WP)**	**Gruppengröße (TN)**
a)	Designing code analyses for large-scale software systems 1	V3 Ü2	75	105	P	30

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

Teilnahmevoraussetzungen der Lehrveranstaltung Designing code analyses for large-scale software systems 1:
Empfohlene Vorkenntnisse
Ein gutes Verständnis von Java und den Prinzipien objektorientierter Programmierung ist hilfreich.

4	**Inhalte:** *Inhalte der Lehrveranstaltung Designing code analyses for large-scale software systems 1:* Statische Codeanalysen dienen dazu, automatisiert Fehler und Schwachstellen im Programmcode aufzufinden. Zu diesem Zwecke suchen sie nach bekannten Fehlermustern. In dieser Vorlesung wird erklärt, wie man solche Codeanalysen entwirft, die inter-prozedural sind, also das komplette Programm betrachten, über die Grenzen einzelner Prozeduren hinweg. Der Entwurf solcher Analysen gestaltet sich deshalb sehr schwierig, weil die Analysen oft Millionen von Programmstatements gleichermaßen präzise aber auch effizient verarbeiten müssen. Es werden außerdem Beispielsanalysen aus dem Bereich der IT-Sicherheit besprochen. Diese Veranstaltung ist Teil einer Kombination DECA 1/2. In DECA 2 werden aktuelle Ansätze aus der Forschung besprochen. Es wird dringend empfohlen zuerst DECA 1 und dann DECA 2 zu belegen. **Behandelte Themen:** • Typsysteme und fluss-insensitive Analysen • Endliche Verbände und Fixpunkte • Intra-prozedurale fluss-sensitive Codeanalysen • Intervallanalyse, Widening und Narrowing • Erstellen von Call-graphen • Pointer-Analyse • Inter-prozedurale Codeanalysen • Context-sensitive Analyse mit dem Call-strings Approach • Context-sensitive Analyse mit dem Functional approach • Value-based Termination, VASCO • Distributive Analysen mit IFDS • Praktische Definitionen von Flussfunktionen • Distributive Analysen mit IDE Während der gesamten Veranstaltung werden Anwendungsbeispiele aus dem Gebiet der Softwaresicherheit diskutiert.
5	**Lernergebnisse und Kompetenzen:** Durch den Besuch erlernen Studierende. . . • wichtige Designentscheidungen beim Entwurf automatisierter Codeanalysen richtig zu treffen • welche Algorithmen für Codeanalysen in welchen Anwendungssituationen am besten geeignet sind • wie man Codeanalysen für reale Probleme aus der IT-Sicherheit entwirft • wie man gängige Begriffliche wie Kontext-, Fluss-, Feld-, und Objekt-Sensitivität korrekt interpretiert • welche Limitierungen statische Codeanalysen aufweisen • welche gängige Codeanalysen für Sicherheitsschwachstellen (OWASP Top 10 etc.) existieren, und wie sich diese mit den vorgestellten Algorithmen umsetzen lassen. **Nichtkognitive Kompetenzen** • Lernkompetenz • Lernmotivation

<table>
<tr><td>6</td><td>Prüfungsleistung:
☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)
<table><tr><th>zu</th><th>Prüfungsform</th><th>Dauer bzw. Umfang</th><th>Gewichtung für die Modulnote</th></tr><tr><td>a)</td><td>Klausur oder mündliche Prüfung</td><td>90-120 min bzw. 40 min</td><td>100%</td></tr></table>Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Prüfungsleistung konkret zu erbringen ist.</td></tr>
<tr><td>7</td><td>Studienleistung, qualifizierte Teilnahme:
<table><tr><th>zu</th><th>Form</th><th>Dauer bzw. Umfang</th><th>SL / QT</th></tr><tr><td>a)</td><td>Schriftliche Übungsaufgaben</td><td></td><td>SL</td></tr></table>Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.</td></tr>
<tr><td>8</td><td>Voraussetzungen für die Teilnahme an Prüfungen:
Bestehen der Studienleistung</td></tr>
<tr><td>9</td><td>Voraussetzungen für die Vergabe von Credits:
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</td></tr>
<tr><td>10</td><td>Gewichtung für Gesamtnote:
Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Masterstudiengang Informatik v3, Masterstudiengang Informatik v4</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Prof. Dr. Eric Bodden</td></tr>
</table>

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Designing code analyses for large-scale software systems 1:* **Methodische Umsetzung** Vorlesung und Gruppenübungen sowie praktische Programmierübungen mit weltweit genutzten Frameworks für die statische Codeanalyse **Lernmaterialien, Literaturangaben** • Thomas Reps, Susan Horwitz, and Mooly Sagiv. 1995. Precise interprocedural dataflow analysis via graph reachability. POPL '95 • Shmuel Sagiv, Thomas W. Reps, and Susan Horwitz. 1995. Precise Interprocedural Dataflow Analysis with Applications to Constant Propagation. TAPSOFT '95 • Akash Lal, Thomas Reps, and Gogul Balakrishnan. 2005. Extended weighted pushdown systems. CAV 2005 • Nomair A. Naeem, Ondrej Lhoták, and Jonathan Rodriguez. 2010. Practical extensions to the IFDS algorithm. CC 2010 • Yannis Smaragdakis, Martin Bravenboer, and Ondrej Lhoták. 2011. Pick your contexts well: understanding object-sensitivity. POPL 2011 • Eric Bodden. 2012. Inter-procedural data-flow analysis with IFDS/IDE and Soot. SOAP 2012 • Rohan Padhye, Uday P. Khedker. Interprocedural Data Flow Analysis in Soot using Value Contexts. SOAP 2013

Erzeugt am 26. Oktober 2021 um 14:34.

Designing code analyses for large-scale software systems 2			
Designing code analyses for large-scale software systems 2			
Modulnummer:	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** Sommersemester
	Studiensemester:	**Dauer (in Sem.):** 1	**Sprache:** en

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	Designing code analyses for large-scale software systems 2	V3 Ü2	75	105	P	30

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

Teilnahmevoraussetzungen der Lehrveranstaltung Designing code analyses for large-scale software systems 2:
Empfohlene Vorkenntnisse
Der vorherige Besuch der Veranstaltung DECA 1 wird dringend empfohlen. Ein gutes Verständnis von Java und/oder C++ und den Prinzipien objektorientierter Programmierung ist hilfreich.

4	**Inhalte:** *Inhalte der Lehrveranstaltung Designing code analyses for large-scale software systems 2:* Statische Codeanalysen dienen dazu, automatisiert Fehler und Schwachstellen im Programmcode aufzufinden. Zu diesem Zwecke suchen sie nach bekannten Fehlermustern. In dieser Vorlesung wird erklärt, wie man solche Codeanalysen entwirft, die inter-prozedural sind, also das komplette Programm betrachten, über die Grenzen einzelner Prozeduren hinweg. Der Entwurf solcher Analysen gestaltet sich deshalb sehr schwierig, weil die Analysen oft Millionen von Programmstatements gleichermaßen präzise aber auch effizient verarbeiten müssen. Es werden außerdem Beispielsanalysen aus dem Bereich der IT-Sicherheit besprochen. Diese Lehrveranstaltung knüpft an an die Veranstaltung DECA 1. In DECA 2 werden vor allem neuartige Konzepte direkt aus der Forschung besprochen, beispielsweise sogenannte demand-driven analyses, welche sich durch eine präzisere und gleichzeitig effizientere Analyse auszeichnen, aber auch Pushdown-Systeme, die eine elegante Modellierung und ebenso schnelle Ausführung von Programmanalysen erlauben. Zu guter letzt erklären wir aktuelle Lösungsansätze zu praktischen Problemen in der statischen Analyse wie beispielsweise der Nutzung von Reflection und nativem Code. **Behandelte Themen:** • Programmanalyse von Software-Produktlinien • Modellierung von Call Stacks und Feldzugriffen mit Pushdown-Systemen • Modellierung von weiterer Analyseinformationen mit Weighted Pushdown Systems • Effizienz- und Präzisionsgewinne durch bedarfsgesteuerte Programmanalyse • Synchronisierte Pushdown-Systeme im Boomerang-Framework • Angewandte Android-Code-Analyse mit FlowDroid • Behandlung von Reflexion mittels TamiFlex • Hybride statische und dynamische Analyse mit Harvester • Lernen von Quell-, Senken- und Sanitizer-Definitionen mit SWAN und SWAN Assist • Erklärbare statische Analyse Während der gesamten Veranstaltung werden Anwendungsbeispiele aus dem Gebiet der Softwaresicherheit diskutiert.
5	**Lernergebnisse und Kompetenzen:** Durch den Besuch erlernen Studierende. . . • wichtige Designentscheidungen beim Entwurf automatisierter Codeanalysen richtig zu treffen • welche Algorithmen für Codeanalysen in welchen Anwendungssituationen am besten geeignet sind • wie man Codeanalysen für reale Probleme aus der IT-Sicherheit entwirft • wie man gängige Begriffliche wie Kontext-, Fluss-, Feld-, und Objekt-Sensitivität korrekt interpretiert • welche Limitierungen statische Codeanalysen aufweisen • welche gängige Codeanalysen für Sicherheitsschwachstellen (OWASP Top 10 etc.) existieren, und wie sich diese mit den vorgestellten Algorithmen umsetzen lassen. **Nichtkognitive Kompetenzen** • Lernkompetenz • Lernmotivation

<table>
<tr><td>6</td><td>Prüfungsleistung:
☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)
<table><tr><th>zu</th><th>Prüfungsform</th><th>Dauer bzw. Umfang</th><th>Gewichtung für die Modulnote</th></tr><tr><td>a)</td><td>Klausur oder mündliche Prüfung</td><td>90-120 min bzw. 40 min</td><td>100%</td></tr></table>Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Prüfungsleistung konkret zu erbringen ist.</td></tr>
<tr><td>7</td><td>Studienleistung, qualifizierte Teilnahme:
<table><tr><th>zu</th><th>Form</th><th>Dauer bzw. Umfang</th><th>SL / QT</th></tr><tr><td>a)</td><td>Schriftliche Übungsaufgaben</td><td></td><td>SL</td></tr></table>Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.</td></tr>
<tr><td>8</td><td>Voraussetzungen für die Teilnahme an Prüfungen:
Bestehen der Studienleistung</td></tr>
<tr><td>9</td><td>Voraussetzungen für die Vergabe von Credits:
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</td></tr>
<tr><td>10</td><td>Gewichtung für Gesamtnote:
Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Masterstudiengang Informatik v3, Masterstudiengang Informatik v4</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Prof. Dr. Eric Bodden</td></tr>
</table>

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Designing code analyses for large-scale software systems 2:* **Methodische Umsetzung** Vorlesung und Gruppenübungen sowie Programmierübungen mittels realer, weltweit genutzter Frameworks für die statische Analyse (bspw. Soot, Phasar, FlowDroid) **Lernmaterialien, Literaturangaben** • Context-, Flow-, and Field-sensitive Data-flow Analysis Using Synchronized Pushdown Systems (Johannes Späth, Karim Ali, Eric Bodden), In Proceedings of the ACM SIGPLAN Symposium on Principles of Programming Languages, pages 48:1–48:29, 3(POPL), 2019. • FlowDroid: Precise Context, Flow, Field, Object-sensitive and Lifecycle-aware Taint Analysis for Android Apps (Steven Arzt, Siegfried Rasthofer, Christian Fritz, Eric Bodden, Alexandre Bartel, Jacques Klein, Yves Le Traon, Damien Octeau, Patrick McDaniel), In Proceedings of the 35th ACM SIGPLAN Conference on Programming Language Design and Implementation, pages 259–269, PLDI '14, ACM, 2014. • Codebase-Adaptive Detection of Security-Relevant Methods (Goran Piskachev, Lisa Nguyen Quang Do, Eric Bodden), In ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA), 2019. • Taming Reflection: Aiding Static Analysis in the Presence of Reflection and Custom Class Loaders (Eric Bodden, Andreas Sewe, Jan Sinschek, Hela Oueslati, Mira Mezini), In ICSE '11: International Conference on Software Engineering, pages 241–250, ACM, 2011.

Erzeugt am 26. Oktober 2021 um 14:34.

Foundations of Cryptography			
Foundations of Cryptography			
Modulnummer:	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** Sommersemester
	Studiensemester:	**Dauer (in Sem.):** 1	**Sprache:** en

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	Foundations of Cryptography	V3 Ü2	75	105	P	25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

Teilnahmevoraussetzungen der Lehrveranstaltung Foundations of Cryptography:
Empfohlene Vorkenntnisse
Basiskenntnisse in IT-Sicherheit und Kryptographie nützlich aber nicht notwendig, Grundkonzepte der Komplexitätstheorie und Wahrscheinlichkeitstheorie

4 **Inhalte:**

Inhalte der Lehrveranstaltung Foundations of Cryptography:
Wichtige Basiskonzepte moderner Kryptographie werden vorgestellt. Hierzu gehören Verschlüsselungsverfahren, digitale Signaturen, Identifikationsprotokolle und Mehrparteienberechnungen werden vorgestellt. In allen Fällen werden formale Sicherheitsdefinitionen vorgestellt und, ausgehend von mathematisch präzisen Annahmen, beweisbar sichere Konstruktionen entwickelt.

- Symmetrische und asymmetrische Verschlüsselung
- Pseudozufallsfunktionen, Einweg-Funktionen, Permutationen mit Falltüren
- Hashfunctions und Authentifizierungscodes
- Digitale Unterschriften, Einmal-Unterschriften und Zufallsorakel
- Identifikationsprotokolle, Σ-Protokolle
- Sichere Mehrparteienberechnungen

5 **Lernergebnisse und Kompetenzen:**

Studierende verstehen wesentliche Konzepte und Methoden moderner Kryptographie. Sie können für Sicherheitsprobleme geeignete kryptographische Techniken auswählen. Sie können Basistechniken der Kryptographie kombinieren und modifizieren, neue Sicherheitskonzepte definieren und die Sicherheit der Konstruktionen bezüglich dieses Definitionen beweisen.
Nichtkognitive Kompetenzen

- Einsatz und Engagement
- Gruppenarbeit
- Lernmotivation
- Schreib- und Lesekompetenz (wissenschaftlich)
- Selbststeuerungskompetenz

6 **Prüfungsleistung:**

☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Prüfungsleistung konkret zu erbringen ist.

7 **Studienleistung, qualifizierte Teilnahme:**

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Schriftliche Übungsaufgaben		SL

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.

8 **Voraussetzungen für die Teilnahme an Prüfungen:**

Bestehen der Studienleistung

9 **Voraussetzungen für die Vergabe von Credits:**

Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.

10 **Gewichtung für Gesamtnote:**

Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).

11 **Verwendung des Moduls in anderen Studiengängen:**

Masterstudiengang Informatik v3, Masterstudiengang Informatik v4

12 **Modulbeauftragte/r:**

Prof. Dr. Johannes Blömer

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Foundations of Cryptography:* **Methodische Umsetzung** Vorlesung mit Übungen, Lesegruppen **Lernmaterialien, Literaturangaben** • Oded Gorldreich, Foundations of Cryptography I,II, • Jonathan Katz, Yehuda Lindell, Introduction to Modern Cryptography • Folien der Vorlesung

Erzeugt am 26. Oktober 2021 um 14:35.

Introduction to Quantum Computation

Introduction to Quantum Computation

Modulnummer:	**Workload (h):**	**Leistungspunkte:**	**Turnus:**
	180	6	Sommersemester
	Studiensemester:	**Dauer (in Sem.):**	**Sprache:**
		1	en

1 **Modulstruktur**

	Lehrveranstaltung	**Lehrform**	**Kontaktzeit (h)**	**Selbststudium (h)**	**Status (P/WP)**	**Gruppengröße (TN)**
a)	Introduction to Quantum Computation	V3 Ü2	75	105	P	40

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

Teilnahmevoraussetzungen der Lehrveranstaltung Introduction to Quantum Computation:
Empfohlene Vorkenntnisse
Lineare Algebra, Algorithmen

4 **Inhalte:**

Inhalte der Lehrveranstaltung Introduction to Quantum Computation:
In dieser Vorlesung werden die Grundlagen von Quanteninformatik und Quanteninformation vorgestellt. Das umfasst eine Einführung in Quantenmechanik, Quantenverschränkung, Quantenalgorithmen, Quantenfehlerkorrektur und Quanteninformation.

- Quantenmechanik
- Quantenverschränkung
- Quantenalgorithmen
- Quantenfehlerkorrektur
- Quanteninformation

5 **Lernergebnisse und Kompetenzen:**

Studierende können:

- die Postulate von Quantenmechanik beschreiben und benutzen,
- die Benutzung von Quantenverschränkung als eine Quelle verstehen,
- grundlegenden Quantenalgorithmen entwickeln und analysieren
- Quantenfehlerkorrektur benutzen,
- grundlegender Quanteninformationskonzepten, wie Entropie, verstehen und benutzen,

Nichtkognitive Kompetenzen

- Lernkompetenz
- Selbststeuerungskompetenz

6 **Prüfungsleistung:**

☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a)	Klausur oder mündliche Prüfung	120-180 min bzw. 40 min	100%

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Prüfungsleistung konkret zu erbringen ist.

7 **Studienleistung, qualifizierte Teilnahme:**

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Schriftliche Übungsaufgaben		SL

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.

8 **Voraussetzungen für die Teilnahme an Prüfungen:**

Bestehen der Studienleistung

9 **Voraussetzungen für die Vergabe von Credits:**

Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.

10 **Gewichtung für Gesamtnote:**

Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).

11 **Verwendung des Moduls in anderen Studiengängen:**

Masterstudiengang Informatik v3, Masterstudiengang Informatik v4

12 **Modulbeauftragte/r:**

Jun. Prof. Dr. Sevag Gharibian

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Introduction to Quantum Computation:* **Methodische Umsetzung** Eine Mischung aus Folien und Tafelanschrieb. Alle wichtigen Konzepte und Techniken werden in Übungen anhand von Beispielen weiter vertieft. **Lernmaterialien, Literaturangaben** • Michael A. Nielsen, Isaac L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press • Vorlesungsfolien, Übungsaufgaben

Erzeugt am 26. Oktober 2021 um 14:35.

Quantum Complexity Theory			
Quantum Complexity Theory			
Modulnummer:	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** Sommersemester
	Studiensemester:	**Dauer (in Sem.):** 1	**Sprache:** en

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	Quantum Complexity Theory	V3 Ü2	75	105	P	20

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

Teilnahmevoraussetzungen der Lehrveranstaltung Quantum Complexity Theory:
Empfohlene Vorkenntnisse
Lineare Algebra, Quanteninformatik

4 **Inhalte:**

Inhalte der Lehrveranstaltung Quantum Complexity Theory:
Diese Vorlesung gibt einen kurzen Überblick über die Grundlagen von Quanteninformatik und wendet sich anschließend der Quantenkomplexitätstheorie zu. Dabei werden sowohl einführende als auch vertiefende Themen behandelt wie die Analoga zu P und NP (bezeichnet als BQP, QCMA, and QMA), Quanten-Erfüllbarkeitsprobleme, Quanten-interaktive Beweise und Tensor-Netzwerke. Begleitend wird semidefinite Programmierung als ein wichtiges Werkzeug eingeführt.

- Komplexitätsklassen BQP, QCMA, QMA
- Quanten-Erfüllbarkeitsprobleme
- Quanten-interaktive Beweise
- Tensor-Netzwerke
- Semidefinite Programmierung

5 **Lernergebnisse und Kompetenzen:**

Studierende können:

- die Postulate von Quantenmechanik beschreiben und benutzen,
- mit Komplexitätsklassen wie BQP und QMA arbeiten,
- QMA-Schwere zeigen,
- Semidefinite Programmierung nutzen,
- Tensor-Netzwerke benutzen, um verschränkte Quantenzustände zu beschreiben

Nichtkognitive Kompetenzen

- Lernkompetenz
- Selbststeuerungskompetenz

6 **Prüfungsleistung:**

☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Prüfungsleistung konkret zu erbringen ist.

7 **Studienleistung, qualifizierte Teilnahme:**

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Schriftliche Übungsaufgaben		SL

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.

8 **Voraussetzungen für die Teilnahme an Prüfungen:**

Bestehen der Studienleistung

9 **Voraussetzungen für die Vergabe von Credits:**

Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.

10 **Gewichtung für Gesamtnote:**

Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).

11 **Verwendung des Moduls in anderen Studiengängen:**

Masterstudiengang Informatik v3, Masterstudiengang Informatik v4

12 **Modulbeauftragte/r:**

Jun. Prof. Dr. Sevag Gharibian

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Quantum Complexity Theory:* **Methodische Umsetzung** Eine Mischung aus Folien und Tafelanschrieb. Alle wichtigen Konzepte und Techniken werden in Übungen anhand von Beispielen weiter vertieft. **Lernmaterialien, Literaturangaben** • Michael A. Nielsen, Isaac L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press • S. Gharibian, Y. Huang, Z. Landau, S. W. Shin, Quantum Hamiltonian Complexity, Foundations and Trends in Theoretical Computer Science • Vorlesungsfolien, Übungsaufgaben

Erzeugt am 26. Oktober 2021 um 14:35.

Real World Crypto Engineering			
Real World Crypto Engineering			
Modulnummer:	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** Wintersemester
	Studiensemester:	**Dauer (in Sem.):** 1	**Sprache:** en

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	Real World Crypto Engineering	V3 Ü2	75	105	P	40

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

Teilnahmevoraussetzungen der Lehrveranstaltung Real World Crypto Engineering:
Empfohlene Vorkenntnisse
Basiskenntnisse in Programmierung, IT-Sicherheit und Kryptographie

4 **Inhalte:**

Inhalte der Lehrveranstaltung Real World Crypto Engineering:
Starke Kryptographie ist nicht immer ausreichend, um die grundlegenden Sicherheitsziele zu schützen. Auch wenn starke kryptographische Algorithmen verwendet werden, kann bei deren Einsatz viel schief gehen. In dieser Vorlesung werden wir auf die wichtigsten Protokolle und kryptographische Schutzmechanismen eingehen (z.B. TLS, SSH, WPA) und werden ihre Basiskonzepte kennenlernen. Anschließend werden wir prominente Angriffe vorstellen, die die gewünschten Sicherheitsziele komplett gebrochen haben. Basierend auf vielen Fällen werden wir lernen, was beim Design und bei der Implementierung von kryptographischen Anwendungen wichtig ist.

5 **Lernergebnisse und Kompetenzen:**

Studierende verfügen nach erfolgreichem Abschluss über ein umfassendes Verständnis der technischen Aspekte von angewandten kryptographischen Algorithmen. Sie haben erkannt, dass Kryptographie alleine nicht ausreicht, um sicherheitstechnische Probleme zu lösen. Sie haben einen Überblick über aktuelle kryptographische Angriffe und wissen, wie man diese praktisch verhindert.
Nichtkognitive Kompetenzen

- Gruppenarbeit
- Schreib- und Lesekompetenz (wissenschaftlich)

<table>
<tr><td>6</td><td>Prüfungsleistung:
☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)<table><tr><th>zu</th><th>Prüfungsform</th><th>Dauer bzw. Umfang</th><th>Gewichtung für die Modulnote</th></tr><tr><td>a)</td><td>Klausur oder mündliche Prüfung</td><td>90-120 min bzw. 40 min</td><td>100%</td></tr></table>Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Prüfungsleistung konkret zu erbringen ist.</td></tr>
<tr><td>7</td><td>Studienleistung, qualifizierte Teilnahme:<table><tr><th>zu</th><th>Form</th><th>Dauer bzw. Umfang</th><th>SL / QT</th></tr><tr><td>a)</td><td>Schriftliche Übungsaufgaben</td><td></td><td>SL</td></tr></table>Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.</td></tr>
<tr><td>8</td><td>Voraussetzungen für die Teilnahme an Prüfungen:
Bestehen der Studienleistung</td></tr>
<tr><td>9</td><td>Voraussetzungen für die Vergabe von Credits:
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</td></tr>
<tr><td>10</td><td>Gewichtung für Gesamtnote:
Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Masterstudiengang Informatik v3, Masterstudiengang Informatik v4</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Prof. Dr.-Ing. Juraj Somorovsky</td></tr>
<tr><td>13</td><td>Sonstige Hinweise:
Hinweise der Lehrveranstaltung Real World Crypto Engineering:
Methodische Umsetzung
Vorlesung mit Übungen
Lernmaterialien, Literaturangaben
Folien der Vorlesung, wissenschaftliche Artikel</td></tr>
</table>

Erzeugt am 26. Oktober 2021 um 14:36.

Web Security			
Web Security			
Modulnummer:	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** Sommersemester
	Studiensemester:	**Dauer (in Sem.):** 1	**Sprache:** en

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	Web Security	V3 Ü2	75	105	P	40

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

Teilnahmevoraussetzungen der Lehrveranstaltung Web Security:
Empfohlene Vorkenntnisse
Kenntnisse in Programmierung, IT-Sicherheit und Basiskenntnisse in Kryptographie

4 **Inhalte:**

Inhalte der Lehrveranstaltung Web Security:
Moderne Webapplikationen und Webservices sind oft vielschichtig und basieren auf unterschiedlichen (oft komplexen) Technologien, die ständig weiterentwickelt werden. Deren Komplexität ist oft der Grund für neuartige Angriffe, die im Web-Bereich täglich zu sehen sind.
In dieser Vorlesung werden wir auf die wichtigsten Technologien eingehen und lernen, worauf man bei der sicheren Web-Entwicklung achten muss. Dabei werden wir prominente und weit verbreitete Angriffe vorstellen und zeigen, wie man die verhindert. Dazu gehören typische Angriffe aus der OWASP Top 10 Liste wie XSS oder SQL Injection bis hin zu Angriffen auf Webservices und Single Sign-On Standards (wie SAML und OpenID Connect). Basierend auf vielen Fällen werden wir lernen, was beim Design und bei der Implementierung von Webapplikationen wichtig ist.

5 **Lernergebnisse und Kompetenzen:**

Studierende verfügen nach erfolgreichem Abschluss über ein umfassendes Verständnis der technischen Aspekte von Webapplikationen, Webservices und diversen Authentizierungsmechanismen. Sie haben erkannt, dass die heutzutage eingesetzten Web-Technologien vielschichtig sind und dass deren Komplexität viele Sicherheitsprobleme mit sich bringt. Studierende haben einen Überblick über aktuelle Web-Angriffe und wissen wie man diese praktisch verhindert.
Nichtkognitive Kompetenzen

- Gruppenarbeit
- Schreib- und Lesekompetenz (wissenschaftlich)

<table>
<tr><td>6</td><td>Prüfungsleistung:
☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)
<table><tr><th>zu</th><th>Prüfungsform</th><th>Dauer bzw. Umfang</th><th>Gewichtung für die Modulnote</th></tr><tr><td>a)</td><td>Klausur oder mündliche Prüfung</td><td>90-120 min bzw. 40 min</td><td>100%</td></tr></table>Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Prüfungsleistung konkret zu erbringen ist.</td></tr>
<tr><td>7</td><td>Studienleistung, qualifizierte Teilnahme:
<table><tr><th>zu</th><th>Form</th><th>Dauer bzw. Umfang</th><th>SL / QT</th></tr><tr><td>a)</td><td>Schriftliche Übungsaufgaben</td><td></td><td>SL</td></tr></table>Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.</td></tr>
<tr><td>8</td><td>Voraussetzungen für die Teilnahme an Prüfungen:
Bestehen der Studienleistung</td></tr>
<tr><td>9</td><td>Voraussetzungen für die Vergabe von Credits:
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</td></tr>
<tr><td>10</td><td>Gewichtung für Gesamtnote:
Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Masterstudiengang Informatik v3, Masterstudiengang Informatik v4</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Prof. Dr.-Ing. Juraj Somorovsky</td></tr>
<tr><td>13</td><td>Sonstige Hinweise:
Hinweise der Lehrveranstaltung Web Security:
Methodische Umsetzung
Vorlesung mit Übungen
Lernmaterialien, Literaturangaben
• Folien der Vorlesung
• Wissenschaftliche Artikel</td></tr>
</table>

Erzeugt am 26. Oktober 2021 um 14:36.

HERAUSGEBER
PRÄSIDIUM DER UNIVERSITÄT PADERBORN
WARBURGER STR. 100
33098 PADERBORN

HTTP://WWW.UNI-PADERBORN.DE

ISSN 2199-2819