Course Change Request

New Course Proposal

Date Submitted: 10/19/18 3:07 pm

Viewing: **MATH 175 : Mathematics of Cryptography: An Introduction**

Last edit: 10/19/18 3:07 pm

Changes proposed by: igriva

Are you completing this form on someone else's behalf?
No

Effective Term: Fall 2019

Subject Code: MATH - Mathematics **Course Number:** 175

Bundled Courses:

Equivalent Courses:

Catalog Title: Mathematics of Cryptography: An Introduction

Banner Title: Mathematics of Cryptography

Will section titles vary by semester? No

Credits: 3

Schedule Type: Lecture w/Lab

Hours of Lecture or Seminar per week: 2h30m

Hours of Lab or Studio per week: 1h15m

Repeatable: May be only taken once for credit, limited to 3 attempts (N3) **Max Allowable Credits:** 3

Default Grade Mode: Undergraduate Regular

Recommended Prerequisite(s): B or better in a calculus course.

Recommended Corequisite(s):

Required Prerequisite(s) / Corequisite(s) (Updates only):

Registrar's Office Use Only - Required Prerequisite(s)/Corequisite(s):

And/Or	(	Course/Test Code	Min Grade/Score	Academic Level	)	Concurrency?

Registration Restrictions (Updates only):

Registrar's Office Use Only - Registration Restrictions:

Field(s) of Study:

Class(es):

Level(s):

Degree(s):

School(s):

Catalog Description: Every day, 143,000 terabytes of data are transferred across the internet, including financial transactions, medical records, and sensitive client data. Half of this traffic is secured through encryption, relying on mathematical algorithms such as the RSA to encode the data in a way that only the recipient can decode.

In Workflow

1. **MATH Chair**
2. **SC Curriculum Committee**
3. SC Associate Dean
4. Assoc Provost-Undergraduate
5. Registrar-Courses
6. Banner

Approval Path

1. 10/19/18 10:08 pm David Walnut (dwalnut): Approved for MATH Chair

In this class, we will see how cryptography works first-hand. We will start with classical ciphers (Atbash and Caesar ciphers) and develop our mathematical techniques and programming abilities until we are able to implement RSA from scratch. Topics covered in the course lead into the following majors: mathematics, computer science, electrical engineering, and cyber security engineering.

Justification:

The goal of the proposed course is to provide an early experience at the interface of advanced mathematical theory and applications in a way that interfaces directly with a wide range of STEM majors and career options. The particular topic of cryptography is particularly approachable: no pre-requisites beyond enthusiasm are expected from the students, but it is possible to arrive at both interesting mathematics and hands-on results within a single semester.

To encourage skill development and ownership of the material, as well as to improve outcomes traditionally under-represented students, the course is taught using the Inquiry Based Learning (IBL) approach: minimal lecturing by the instructor, with all knowledge developed by students through self-paced groupwork and disseminated among students through informal group presentations. The course format and materials follow the course ``Introduction to Cryptology (Math175)'' at the University of Michigan, taught and further developed by Lukyanenko over three years as a postdoc. The course is quite popular at the University of Michigan, and attracts a wide audience of students majoring in topics including computer science, performing arts, and business. In its current test implementation at GMU as Math493-002, the course has filled the allocated room with 11 enthusiastic students with a variety of majors, who now appear substantially more enthusiastic about mathematics courses, seminar talks, and undergraduate research opportunities.

The course itself consists of two components: classroom and lab. In the classroom (two 1h15m sessions per week), the students work in groups to develop the mathematics behind modern cryptography, guided only by their own intuition and the instructor's questions. They start by examining examples of the oldest known cyphers: the Atbash cypher reverses the alphabet (the ``a''s in the text become ``z''s, ``b''s become ``y''s and so on), while the Caesar cipher used by Julius Caesar shifts text by 3 letters (so ``hello'' becomes ``khoor''). This leads to mathematical notions: negation and addition modulo 26, respectively. Understanding these concepts thoroughly leads to the ability to decode simpler ciphers and the develop new ones. In parallel, the students work in a computer lab (one 1h15m session per week) to develop the programming skills necessary to implement the mathematical algorithms invented in the classroom, validating and motivating the theoretical development. Starting from the basics of mathematics and programming, the course develops towards an understanding of the most prevalent modern cryptographic algorithm: the RSA. The students prove Fermat's Little Theorem that lies at the heart of the algorithm, and use their own implementation of RSA to exchange secret messages in the classroom.

Does this course cover material which crosses into another department? No

Learning Outcomes:

Because of the inquiry-based and interdisciplinary nature of the course, it accomplishes a wide range of outcomes. Content-wise, the students learn the following topics at the introductory level:
1. Fundamentals of mathematics (Peano axioms and definition of integer arithmetic),
2. Number theory (modular arithmetic, Euclidean algorithm, primality, multiplicative inverses, Fermat's Little Theorem),
3. Cryptography (classical ciphers, affine ciphers, statistical methods, public key exchange, RSA),
4. Programming (logic, loops, text analysis, functional programming).

The following skills are learned in the course:
1. Mathematical thinking: proofs, axioms, and definitions, including an informal algorithmic interpretation of induction,
2. Programming: teaching themselves a new language, writing code, developing algorithms, and debugging broken code.
3. Technical communication: effective groupwork, presenting and critiquing theoretical progress, and scientific writing.

Evaluation of the skills is performed through three methods:
1. Participation grade,
2. Weekly homework assignments,
3. Monthly submission of textbook-style writeups of developed theory,
4. Final submission of a ``final textbook'' including corrections to previous submissions,
5. Weekly lab worksheet submissions.

Attach Syllabus syllabus.pdf

Additional Attachments	Cryptography Mid-Semester Feedback.pdf
Staffing:	Anton Lukyanenko Rebecca R.G. Geir Agnarsson Neil Epstein Sean Lawton
Relationship to Existing Programs:	The course is an elective course that is intended to stimulate and inform students' interest in the mathematics of cryptography, cyber security, and computer science.
Relationship to Existing Courses:	The course provides a thorough introduction to proof methods, which are used in all higher-level mathematics courses. Direct follow-up courses in the mathematics department include number theory (Math301) and abstract algebra(Math321).

Additional Comments:	The course is taught this semester as Math493, a topics course, but will be taught as Math175 in the future, if approved. The attached file is a recent mid-semester survey of the Fall 2018 students, indicating a high degree of enthusiasm for the course.
Reviewer Comments	

Key: 16076

Math 493
Mathematics of Cryptography: an Introduction
George Mason University, Fall 2018
`http://lukyanenko.net/teaching/2018/493/`

Instructor:
Anton Lukyanenko
alukyane@gmu.edu

Office Hours:
Thursdays 4:30-5:55pm & by appointment
4113 Exploratory Hall

Class:
Mondays-Wednesdays 10:30-11:45
4301 Exploratory Hall

Thursdays 10:30-11:45
4307 Exploratory Hall

Course content. Every day, 143,000 terabytes of data are transferred across the internet, including financial transactions, medical records, and sensitive client data.

Half of this traffic is secured through encryption, relying on mathematical algorithms such as the RSA to encode the data in a way that only the recipient can decode.

In this class, we will see how cryptography works first-hand. We will start with classical ciphers (Atbash and Caesar ciphers) and develop our mathematical techniques and programming abilities until we are able to implement RSA from scratch.

Topics covered in the course lead into the following majors: mathematics, computer science, electrical engineering, and cyber security engineering.

Workflow and assignments. The course is teamwork-based, with discussion guided by worksheets (in the classroom) and labs (in the computer lab).

In the classroom, we will develop mathematical theory in small groups. Once enough problems on a worksheet are solved, a group will present them on the board. Once the entire worksheet is completed (however long that takes), it is written up carefully at home and becomes part of the final textbook. The homework (supplementary exercises) listed at the end of each worksheet is due a week after the worksheet is completed.

In the computer lab, we will implement the theory we develop in class and explore some additional topics. Computer lab assignments are turned in once they are completed and do not contain any further homework.

Grade breakdown. For the final grade, assignments will be weighed as follows:

Homework: 35%
Class participation: 25%
Labs: 20%
Final textbook: 20%

Letter grades will be based on the usual breakdown (90-93.3 for A-, 93.4-96.6 for A, 96.7-100 for A+, etc).

Worksheets There are 13 worksheets, which build up the mathematical theory we need to verify that the RSA algorithm works as desired:

1. Codes
2. Numbers
3. Modular addition
4. Shift ciphers
5. Remainders
6. Modular multiplication
7. Multiplicative inverses
8. Affine ciphers
9. The Euclidean algorithm
10. The extended Euclidean algorithm
11. RSA encryption
12. Prime numbers
13. Fermat's and Euler's Little Theorems

Labs There 8 lab assignments, which implement the things we develop in the classroom and explore some additional topics. All labwork uses Mathematica, and you are encouraged to search online and in documentation for ways to solve the problems. However, no additional software is allowed (no python, perl, etc.).

1. Cracking codes
2. Modular arithmetic
3. Frequency analysis
4. Kid Krypto
5. Base-26
6. Recursive functions
7. MyPowerMod
8. Putting RSA Together

Final textbook. At the end of the course (on **December 19**), students will turn in a 'textbook' consisting of their solutions to all problems from in-class worksheets. On the following dates, students will submit writeups of all worksheets (1) not submitted previously and (2) completed by the Wednesday prior to the due date:

September 21 October 5 October 26 November 16

Worksheets will be graded on correctness and clarity of exposition. Students are encouraged to type up their solutions in LaTeX for easier editing, but neat handwritten copies will also be accepted.

Each of the above blocks will count 5% toward the final textbook grade. The final project, complete solutions to all worksheets that incorporate previous comments, counts the remaining 80% and is due in class on **December 19**.

Legalities and resources. Hopefully, everyone in the class will have a good time and learn a lot. The policies below exist to encourage these two goals.

Participation and attendance. The participation grade will include attendance, effort, and collegiality. Both contributing to the team and helping others contribute are essential to the course.

In particular, coming to class is extremely important in this course for both the student and their team.

Serial Absenteeism Clause. A student is allowed at most three unexcused absences from class. The instructor will decide what is an acceptable absence on a case-by-case basis, and all absences due to illness require a note from University Health Service. For every unexcused absence beyond the third, the student's final grade will be reduced by one letter grade. For example, a student with four unexcused absences and a final grade of A will receive a B, and a student with a final grade of A and five unexcused absences will receive a C.

Groupwork vs. cheating. Groupwork is critical to the format of the class, and students are encouraged to work in groups on all homework. That said, all submitted work must be the student's own.

Any copying (especially verbatim) is unacceptable, and will result in a zero for the entire assignment. A second instance of cheating on homework will result in automatic failure of the course. Late homework will not be accepted, except in grave emergencies, and will count zero.

Conducive environment. A pleasant and accommodating environment for all students is critical for learning. In particular, the instructor is happy to provide individualized support during both reguarly scheduled and additional office hours; while the university provides support services for a variety of ongoing conditions and emergencies.

Any violations to the above standard are taken very seriously by the university. In particular, any cases of discrimination, harassment, or violence involving students are investigated by the Dean of Students and can lead to expulsion from the university and/or criminal charges.

Resources. The following groups exist to support student learning, with both academic and non-academic issues, so don't hesitate to contact them:

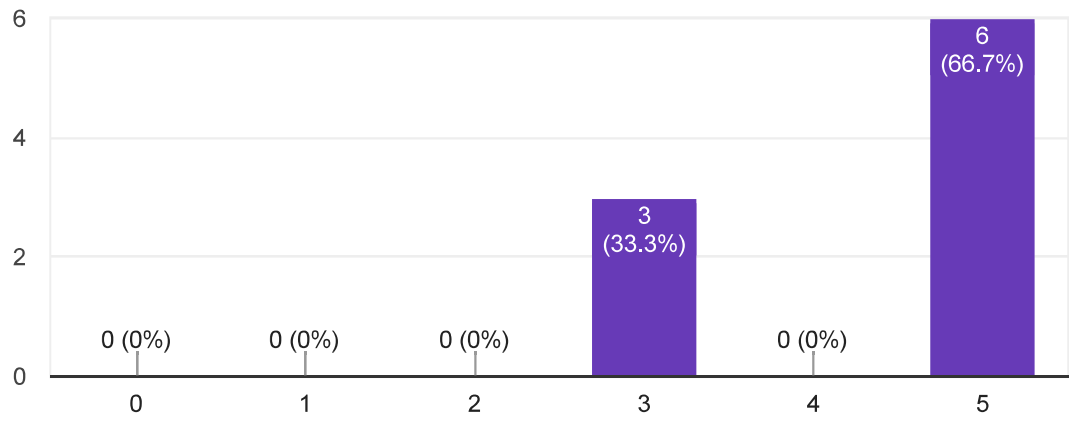
- College of Science Adivisng: `https://cos.gmu.edu/uaa/advising/`
- Department of Mathematics Major Information: `http://math.gmu.edu/undergrad-student-resources.php`
- Department of Mathematics Advising and Contacts: `http://math.gmu.edu/contacts.php`
- Disability Services: `https://ds.gmu.edu/`
- Counseling and Psychology Services `https://caps.gmu.edu/`

Cryptography Mid-Semester Feedback

9 responses

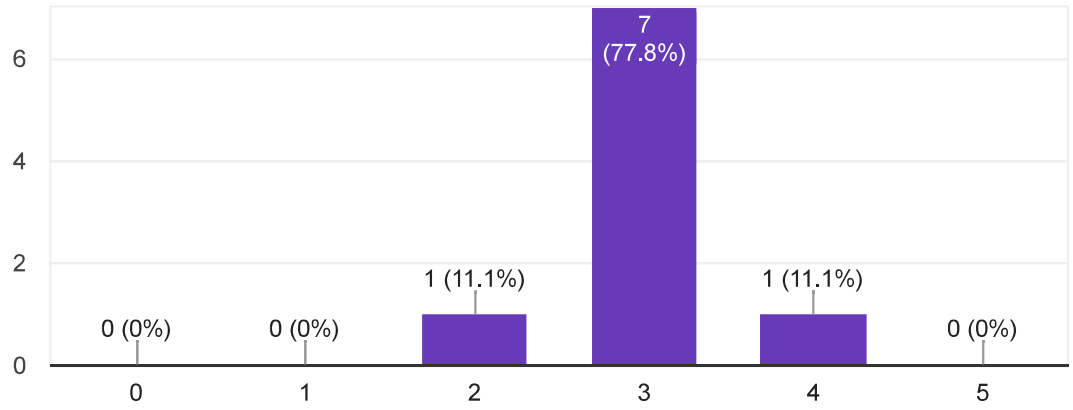
Is your understanding of course content improving?

9 responses



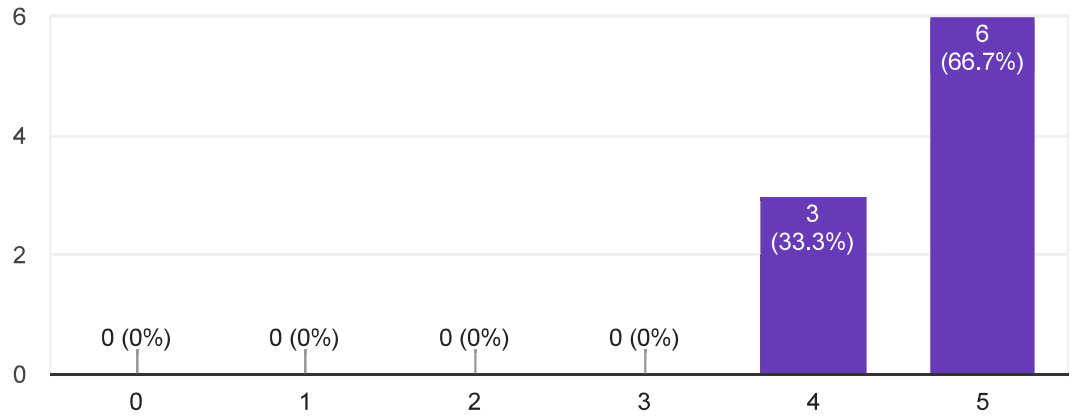
How's the workload?

9 responses



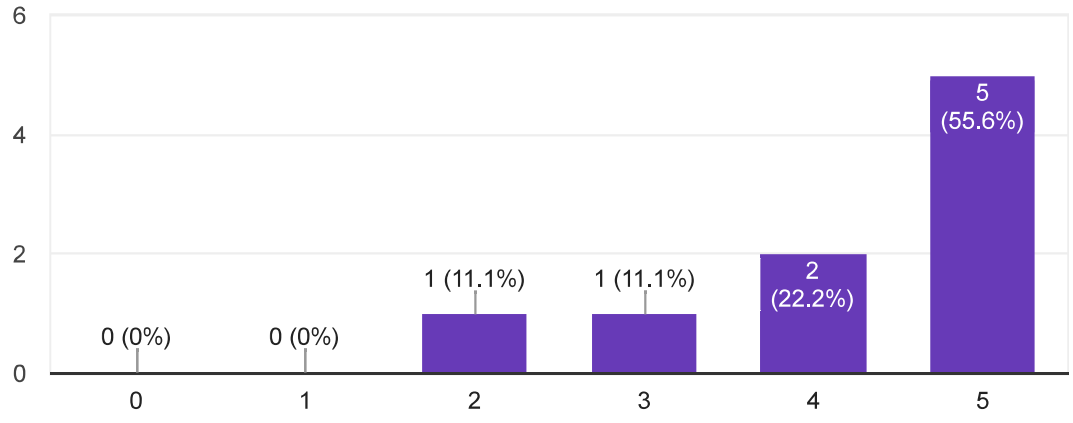
Was it a good idea for you to sign up for this class?

9 responses



Do you have a better understanding of what mathematics is, and whether you are interested in taking more math classes?

9 responses



What can the instructor do to improve the class?

8 responses

Possibly a little more lecture so we are not simply going from worksheet to worksheet

No complaints, class has been awsome so far!

I know this is the part that is often most helpful, but I'm not sure what could be improved upon! You're doing great, Anton!

Nothing comes to mind yet. I like what you're doing.

I cant think of anything he is doing a good job!

Remember that this class was supposed to be for students with very little CS/proofs background, the students with a lot of previous knowledge and who are more eager to talk may make the instructor more likely to not explain the concepts as thoroughly for the students who are not at that level yet.

I feel that it's hard to get a grasp on the proofs explained in class. I wish we had readings that would help us understand the proofs better. I also wish we had a lab that explained how to use LateX. I also wish we had better background in cryptography and how what we do relates to cryptography

Nada

What can other students do to improve the class?

6 responses

Spending more time on out of class assignments. They may seam like the sorts of assignments that can be completed the day before, but I try to start them as soon as they're due date is announced. This gives ample time to reflect and ensure full understanding of the material.

N/A

They could be a little bit more curious when other students are presenting.

One student in particular should make more of an effort to pay attention and follow along when we're going through the exercises on the whiteboard, instead of waiting until the end to say that they didn't understand anything that was just said.

I wish we had better communication among one another outside of the classroom. An active online forum would be a great way to spread ideas and concepts. Sort of like piazza in CS 112.

Be quiet

Other comments about the class so far?

7 responses

The workload is the perfect amount, like the problem solving style class because makes class more interesting and engaging

Seriously ideal instruction. It's obvious that the instructor has taught this course before and the structure of the class is very intentional. I'm SO glad I'm taking this vs 125.

I like the class a bunch! I wish there was a part 2.

I really enjoy it and I would highly recommend it to everyone.

The instructions are very unclear on what is expected in the homework and textbook write ups. It would be better if the professor took some time to thoroughly explain what exactly he is looking for, giving examples of good textbooks and bad ones.

This class is amazing, sure it has its flaws. But it is the first time the class is being offered so I get that it may have a few bugs.

Nada

For next fall, what should be taken into account for scheduling the class? What other courses should we avoid overlapping with?

8 responses

It might be hard to get students and fit their schedule if the lab is on Thursday since many classes are Monday/Wednesday or Tuesday/Thursday maybe use lab as every other Wednesday class

I had to drop a differential equations section to take this course, so avoid core classes, especially math/cs/engineering core

Cs112

N/A

I feel like this course covers what is learned in discreet math. I feel like it is a more fun and interesting way to learn the concepts.

Make sure to include the Thursday lab time in to the official GMU scheduling software. I did not end up having a conflict, but I didn't realize that I could have had a conflict when scheduling my other classes because the lab time didn't show up in my scheduling tool.

I feel like the pre-req's of this class should be calc, and Cs 112. After that I feel that the student will be in good shape to keep up with the class. I feel very out-gunned in this class in terms of skill. It seems everyone can code, and have a good familiarity with cryptography. The skill-level of students is not as even as it is in other classes.

Everything