THE UNIVERSITY OF SYDNEY
Semester 2, 2017

Information Sheet for **MATH2068/2988 Number Theory and Cryptography**

Websites: It is important that you check the following webpages regularly.

Intermediate Mathematics webpage: `http://www.maths.usyd.edu.au/u/UG/IM/`
MATH2068/2988 webpage: `http://www.maths.usyd.edu.au/u/UG/IM/MATH2068/`

(For ease of updating, all information about the advanced unit MATH2988 will be placed on the MATH2068 page, which thus becomes the combined page for both units.) Both pages may be accessed through the Learning Management System (Blackboard):

`https://elearning.sydney.edu.au`

There will also be a question-and-answer forum for MATH2068/2988 on the Ed site:

`https://edstem.com.au/login`

Important announcements relating to Intermediate Mathematics as a whole will be posted on the Intermediate Mathematics page. On the MATH2068/2988 page you will find online resources as described below and other useful links. Announcements regarding assessment tasks will be made on this page at various times throughout the semester, as well as on the Ed forum.

Lectures: The lecturer for this unit is Associate Professor Dzmitry Badziahin. Units MATH2068 and MATH2988 share the same three lectures per week, at the following times and locations.

Time	**Location**
1pm Mon	Carslaw 159
2pm Tue	Carslaw 157
2pm Wed	Carlsaw 159

Lectures run for 13 weeks, from the first lecture on Monday 31 July to the last lecture on Wednesday 1 November. Note that there will be no lectures during the mid-semester break or on Monday 2 October (public holiday). Lectures will be recorded by the Echo system, and the recordings will be available through the Learning Management System (Blackboard) website. Handwritten lecture notes (and occasional presentation slides) will be posted on the MATH2068/2988 webpage.

Consultation times: A/Prof Badziahin will be available for consultation from 3pm to 4pm on Wednesday, Weeks 1–13. His office is Carslaw 634.

Tutorials: Tutorials (one per week) run from Week 2 to Week 13. You should attend the tutorial given on your personal timetable; MATH2988 students have a separate tutorial. Roughly, the tutorial in Week n relates to the material from the lectures in Week $n-1$. The tutorial classes in Week 9 will be devoted to the quiz (see below). The exercise sheets for all other weeks' tutorials will be available on the MATH2068/2988 webpage, for you to print out beforehand or access online during the tutorial as you prefer. Solutions to tutorial exercises for Week n will usually be posted on the afternoon of the Thursday of Week n, after all the tutorials.

Computer laboratory sessions: Computer laboratory sessions (one per week) run from Week 1 to Week 12. You should attend the lab given on your personal timetable; MATH2068 and MATH2988 share the same labs. The sessions in Weeks 6 and 12 can be used to do the computer sections of Assignments 1 and 2 (see below). The exercise sheets for all other weeks' computer labs will be available on the MATH2068/2988 webpage, for you to print out beforehand or access online during the lab session as you prefer. Sample log files with comments (which take the place of solutions) will be posted beforehand as well. It is possible to download the relevant software to your own computer so that you can do the lab exercises and computer assignments in your own time; see the MATH2068/2988 webpage for instructions.

Course textbook: R. Howlett, *Number Theory and Cryptography*, School of Mathematics and Statistics, University of Sydney, 2017. Although lecture notes will be posted, you will probably need a copy of the textbook, which is available from Kopystop. The changes from previous years' editions are minimal, so second-hand copies will be fine.

Reference book: Kenneth H. Rosen, *Elementary Number Theory and its Applications*, 6th Edition, Pearson, 2011. You do not need to buy this, but it may be useful for those who want an additional reference.

Assessment: Your final mark for this unit of study will be calculated as follows:

70%: Exam at end of Semester 2.
10%: Quiz mark (using the better mark principle).
10%: Assignment 1 mark.
10%: Assignment 2 mark.

The *better mark principle* means that the quiz mark counts if and only if it is better than or equal to your exam mark. If your quiz mark is less than your exam mark, the exam mark will be used for that portion of your assessment instead. The assignment marks count for 10% regardless of whether they are better than your exam mark or not. The quiz will be the same for MATH2068 and MATH2988 students. The assignments and the exam will have some questions in common, and some questions only for MATH2068 students or only for MATH2988 students. Final marks are returned within one of the following bands:

> **High Distinction (HD), 85–100:** representing complete or close to complete mastery of the material; **Distinction (D), 75–84:** representing excellence, but substantially less than complete mastery; **Credit (CR), 65–74:** representing a creditable performance that goes beyond routine knowledge and understanding, but less than excellence; **Pass (P), 50–64:** representing at least routine knowledge and understanding over a spectrum of topics and important ideas and concepts in the course.

A student with a passing or higher grade should be well prepared to undertake further studies in mathematics which are dependent on this unit of study.

Examination: There is one examination of 2 hours' duration during the examination period at the end of Semester 2, which will test the learning outcomes attained in lectures, tutorials and computer labs. University-approved non-programmable calculators may be used. Further information about the exam will be posted later on the webpage.

Quiz: A quiz will be held during tutorial classes in Week 9. You must sit for the quiz during the tutorial in which you are enrolled, unless you have a Permission Slip from the Student Services Office, issued only for verifiable reasons. Otherwise, your quiz mark may not be recorded. The quiz will consist of short-answer questions testing your understanding of basic concepts and computational methods from the lectures and tutorials in Weeks 1–7. University-approved non-programmable calculators may be used. A sample quiz and further information about the quiz will be posted later on the webpage.

Assignments: There are two assignments, which must be submitted electronically in Turnitin (an internet-based plagiarism-prevention service), via the Learning Management System (Blackboard) website, by the deadline. Note that a submission will not be marked if it is illegible, sideways or upside down. It is your responsibility to check your submission receipt (which will be automatically emailed to you) to ensure that your assignment has been submitted correctly. The assignments will have a theoretical part testing your understanding of the number theory developed in lectures and tutorials (including your ability to write correct proofs) and a computer part testing your understanding of the cryptographic concepts and methods developed in lectures and computer labs. The assignments, including more detailed submission instructions, will be released on the webpage according to the schedule below.

Assessment and feedback schedule:

Task	Available	Deadline/date	Latest extension*	Feedback
Assignment 1	Fri 25 Aug	11:59pm Thu 7 Sep	11:59pm Thu 14 Sep	9am Mon 18 Sep
Quiz		4–5 Sep (Week 9)		11–12 Oct (Week 10)
Assignment 2	Fri 13 Oct	11:59pm Thu 26 Oct	11:59pm Thu 2 Nov	9am Mon 6 Nov

*Extensions for assignments are only possible for students registered with Disability Services or applying for Special Consideration or Special Arrangements.

Special consideration and special arrangements: While studying at the University of Sydney, you may need to apply for special consideration or special arrangements as follows:

> Special consideration may be granted to students where well-attested illness, injury, or misadventure occurs to them (or someone they have carer's responsibility for) during the semester or the exam period. Special arrangements may be granted for essential community commitments. Further information on eligibility, document requirements, and how to apply is available at `http://sydney.edu.au/science/cstudent/ug/forms.shtml#special_consideration`. Applications must be made using the University's formal application process.

You should *not* submit an application of either type

- if you are absent from a lecture, tutorial or computer lab session, since there is no assessment associated with the missed class, or
- if you miss the quiz, since the better mark principle applies.

The assessment category for the assignments is "Submitted Work".

Simple extensions: Part 14 of the University Coursework Policy allows students to apply for a Simple Extension of up to 2 working days on a (non-examination) assessment task. Any request must be made by email to the lecturer, detailing the reason for the request, the students name, student identification number and the unit of study code. Unit coordinators are not obliged to grant Simple Extensions and the decision is not subject to appeal; in this unit, Simple Extensions for the assignments will only be granted on rare occasions when the lecturer is satisfied that it is appropriate. Special Consideration, as described above, is the usual method for a student to seek consideration for an assessment task such as an assignment.

Where to go for help: For help with mathematics, you can post a question on the Ed forum (anonymously from other students if you prefer), ask your tutor during a tutorial or computer lab, consult the lecturer in his consultation time (see above), or email `dzmitry.badziahin@sydney.edu.au` . For administrative questions, first check carefully whether the answers are on this information sheet or on the MATH2068/2988 webpage; if not, ask on the Ed forum or (if the question is specific to your situation) ask at the Student Services Office (Carslaw 520) or email `MATH2068@maths.usyd.edu.au` . Ensure that any emails that you send contain your name and SID, because anonymous emails will be ignored. If your email includes questions that other students would benefit from seeing the answers to, you may be asked to post them on the Ed forum so that they can be answered there.

Objectives: The objectives of this unit are to:

- introduce basic concepts of number theory, such as primes and prime factorization, modular arithmetic, divisors and multiplicative functions, powers and discrete logarithms;
- explore standard proof techniques in number theory, such as induction and proof by contradiction;
- describe standard algorithms for computations in number theory, such as the Euclidean Algorithm and Pollard Rho algorithms;
- introduce the computer algebra package MAGMA and some of its number-theoretic functions;
- introduce basic concepts of cryptography, such as ciphering and deciphering, public and secret keys;
- describe classical cryptosystems such as substitution, transposition and block transposition ciphers;
- explore statistical attacks on classical cryptosystems;
- describe number-theoretic cryptosystems such as RSA, Elgamal, Diffie–Hellman, Rabin's cryptosystem;
- introduce basic concepts of computational complexity, and use them to understand the relative effectiveness of cryptosystems;
- illustrate more difficult results in number theory.

Outcomes: Students who successfully complete this unit should be able to:

- understand and use the basic terminology of number theory and cryptography;
- carry out simple number-theoretic computations either with a calculator or using MAGMA;
- apply standard number-theoretic algorithms;
- understand and use some classical and number-theoretic cryptosystems;
- apply standard methods to attack some classical cryptosystems;
- understand (see below) the theory underlying number-theoretic algorithms and cryptosystems, including the general properties of primes, prime factorization, modular arithmetic, divisors and multiplicative functions, powers and discrete logarithms.

MATH2068 vs MATH2988: The main distinction in the expected outcomes between the mainstream unit MATH2068 and the advanced unit MATH2988 is in the depth of understanding of the underlying theory, both streams being expected to be familiar with the computational methods and algorithms described in the unit. MATH2068 students should be able to reproduce the proofs of the easier theorems in lectures and produce their own proofs of results at a similar level. MATH2988 students should be able to reproduce the proofs of the more difficult theorems in lectures (unless explicitly stated), produce their own proofs of results at a similar level, and devise their own algorithms to solve simple computational problems. Exercises in tutorials and computer labs are starred to indicate the level of difficulty.

Tentative week-by-week outline:

Week	Topics
1	Introduction, divisibility, prime and composite numbers Greatest common divisors, division algorithm, (extended) Euclidean algorithm Factorization: trial division and Fermat's method
2	Congruence notation, complete systems, reduced systems Fundamental Theorem of Arithmetic Inverses, powers and order in modular arithmetic
3	Basic concepts of cryptography, classical cryptosystems Statistical attacks on classical cryptosystems Euler–Fermat Theorem, Fermat's Little Theorem
4	Relating congruences with different moduli Chinese Remainder Theorem, more on powers (Not for assessment) The Data Encryption Standard cryptosystem
5	Multiplicative functions, Euler's phi function Sum and number of divisors, perfect numbers
6	Relating different multiplicative functions Möbius inversion formula The RSA public key cryptosystem
7	Computational complexity, bit operations Big-O notation, polynomial-time algorithms Computational complexity of Euclidean and power algorithms
8	Pollard's Rho factorization algorithm Polynomial congruences Powers and primitive roots modulo a prime
9	Diffie–Hellman key exchange protocol, Elgamal cryptosystem Safe primes, discrete logarithms Applications of primitive roots
10	Modular Lagrange interpolation formula and secret sharing Baby-step/giant-step and Pohlig–Hellman algorithm for discrete logarithms
11	More on computing discrete logarithms Square roots modulo a prime Rabin's public key cryptosystem
12	(Not for assessment) Quadratic reciprocity theorem and other topics
13	Revision