New York University Tandon School of Engineering

Computer Science Department

Course Outline for CS-GY 6903-I and CS-UY 4783-A

Semester: **Fall 2018**

Instructor: Giovanni Di Crescenzo

To contact instructor: gdc258@nyu.edu

Lecture day, times and room: Tuesday, 6:00pm - 8:30pm in RH 331 (there may be a few exceptions; for instance, Lecture 4 will most likely be held online via WebEx on Sep 26 or 24 instead of Sep 25; any lectures held online or at a different day or time will be recorded and the recording will be made available to all students)

Course Pre-requisites:

(CS-UY 2134 or CS-UY 1134) and (CS-UY 2124 or CS-UY 1124) (C- or better).

The following technical pre-requisites are expected from students attending this class:

1) some expertise in a programming language, like C, Python, etc.
2) some mathematical maturity, in terms of understanding and working with mathematical definitions, concepts, and proofs, and elementary notions of logic, set theory, number theory, probability and statistics; and
3) knowledge of basic algorithm analysis and complexity theory, as obtained from a graduate algorithms class.

Although relevant notions about logic, set theory, number theory, probability, statistics, algorithms, and complexity theory will be provided as part of the class lectures, students are still recommended to take this class after taking a graduate Algorithms class. If not possible, students are strongly encouraged to fill any relevant gaps using web links provided during class lectures as well as e-mail, online or in-person interactions with the instructor. A quick reading of the web content at whichever applicable among the following web links, is strongly recommended:

1) Propositional Logic: http://en.wikipedia.org/wiki/Propositional_calculus (only "Basic concepts" section), http://en.wikipedia.org/wiki/Truth_table, http://en.wikipedia.org/wiki/Contraposition
2) Set Theory: http://en.wikipedia.org/wiki/Algebra_of_sets
3) Probability: https://en.wikipedia.org/wiki/Probability#Mathematical_treatment
4) Algorithms: http://en.wikipedia.org/wiki/Analysis_of_algorithms, http://en.wikipedia.org/wiki/Big_O_notation
5) Complexity Theory: http://en.wikipedia.org/wiki/P_versus_NP_problem

Course Description

The last 40+ years have witnessed a revolution in the area of Cryptography, bringing real-life security problems to the attention of a vast research community. This revolution created Modern Cryptography, where researchers started rigorously treating and solving several problems that only a few years before were unknown or seemed impossible to solve. Today Modern Cryptography is a well-established mathematical discipline, with strong connections to several older disciplines such as

Complexity Theory, Information Theory, Combinatorics, Number Theory, and Coding Theory, and several applications to real-life problems. This Applied Cryptography class offers a comprehensive introduction to Modern Cryptography, and, specifically, its main problems, formalisms, solutions, and open questions, with a heavy focus on application aspects, including case studies for real-life uses of Modern Cryptography solutions.

Course Objectives and Outcomes

The course will provide students with the opportunity to:

1. Learn the main areas of Modern Cryptography, including their main problem statements and the rigorous mathematical approaches used to formalize them
2. Learn and describe how various cryptographic algorithms and protocols work, and the main techniques used in them
3. Evaluate functionality, security and performance properties of cryptography methods used as components of complex security solutions
4. Analyze the impact of errors or different designs of cryptography algorithms and protocols
5. Describe the applications of cryptography algorithms and protocols to real-life problems and many implementation issues in developing these solutions.

At the end of the class the diligent student is also expected to be ready to initiate an advanced study or a research/development project on problems in the area, understand and use cryptographic software tools, and select known cryptographic solutions (e.g., algorithms, protocols, key lengths, etc.) for a desired cryptography application.

Course Structure

Except for textbooks, the course's **technical material**, including lecture slides and explanation videos, is posted online and available so that students can read or listen to it whenever and as many times it is desired or needed. This material is divided into 12 weeks, and contains, for each of the 12 weeks,

1) a **lecture document**; i.e., a PDF file containing between 25 and 40 PowerPoint slides, with pointers to required or recommended textbooks, posted online by Monday of the appropriate week (usually, many weeks in advance); and
2) a **lecture video;** i.e., a video recording of a previously recorded explanation of (an older version of) the lecture slides, posted together with the lecture document; this recording is usually superseded by the live presentation performed during the semester, and is thus only recommended for some specific needs (e.g., studying ahead, need for an alternative explanation of the same concept, etc.).

The course's **interaction with the instructor and (if any) TAs** is structured as follows. If you are an onsite student, your onsite meeting is at the designated NYU Tandon School of Engineering classroom (see top of first page); if you are an online student, your online meeting is via NYU Classes (or you can attend the onsite meeting in person if you are in the area). On any week, you can:

1) attend a weekly **meeting,** usually on Tuesday 6pm EST (but see class schedule below), which will be one of the following:
   - **a.** a **lecture summary, discussion and elaboration**, which assumes the students have already listened to the lecture video, and briefly summarizes the lecture slides, discussing and elaborating on some key topics, especially when they relate to the current homework; or
   - **b.** (most likely:) a detailed **presentation of the current lecture**;

   attending meetings is not mandatory but may improve your class grade; students are strongly encouraged to attend and actively participate by posing their questions;

2) attend an **office hour with the instructor** (usually, right before and after the weekly meeting; say, Tuesday 5:30pm-6pm EST and 8:30pm-9pm EST); in these office hours the instructor will address any questions on the lectures, homework, projects, extra credit, midterm and final; attending instructor office hours is not mandatory but students are strongly encouraged to attend and actively participate by posing their questions;
3) attend an **office hour with the TA, if any** (usually, on Wednesday and/or Thursday 8pm-9pm EST but check the class website for updated hours); in these office hours the TA will address any questions on the lectures, homework, projects, extra credit, midterm and final; attending TA office hours is not mandatory but students are strongly encouraged to attend and actively participate by posing their questions.

Exact days and times may vary, possibly based on students' feedback.
The instructor will send a weekly email to remind students of their expected duties for the week, including any updates to class duties and/or schedule. Students are strongly encouraged to follow the recommended schedule of class duties.

The technical course material is divided into 12 lectures, whose content is detailed below.

Lecture 1: History of cryptography, some background in probability and algorithms, classical cryptography (shift cipher, monoalphabetic substitution cipher, polyalphabetic substitution cipher), encryption with perfect secrecy, one-time pad

Lecture 2: Some background in algorithms and complexity theory, modern cryptography principles, one-way functions, trapdoor functions, hard-core bits, construction of a public-key cryptosystem based on general cryptographic primitives, implementation aspects of computational efficiency and hardness

Lecture 3: Algorithmic number theory, number theory and cryptographic assumptions, Reductions, proofs by reductions, number theory candidates for cryptographic primitives (e.g., factoring and related problems), public-key cryptosystems from number theory problems, key lengths in implemented public-key cryptosystems; brief discussion of quantum computing

Lecture 4: Randomness and pseudo-randomness, pseudo-random generators, functions and permutations

Lecture 5: Symmetric encryption: introduction, security notions, symmetric encryption schemes based on pseudo-randomness primitives, security proofs, fundamental concepts

Lecture 6: Symmetric encryption: block ciphers (e.g., DES, Triple-DES, AES), substitution/permutation networks, Feistel networks, modes of operations (e.g., ECB, CBC, OFB, Counter), cryptanalysis attacks (e.g., exhaustive, linear, differential, meet-in-the-middle attack), key lengths

Lecture 7: Message authentication: introduction, notion and schemes (e.g., CBC-MAC), collision-resistant hashing (MD5, SHA-1, SHA-2, SHA-3, HMAC, Merkle-Hellman), CCA security for symmetric encryption, simultaneous message confidentiality and message integrity, application case study 1: password-based secure computer access

Lecture 8: More number theory candidates for cryptographic primitives (e.g., discrete logarithms, brief discussion of related problems including elliptic curves). Asymmetric encryption: comparison with symmetric encryption, definitions, constructions (e.g., RSA variants, El Gamal), hybrid encryption

Lecture 9: Asymmetric encryption: malleable and homomorphic encryption notion and schemes (e.g., Paillier, brief discussion of various schemes, including Gentry's), additional schemes achieving various security notions in various models (e.g., Cramer-Shoup), identity-based encryption, timed-release encryption

Lecture 10: Digital Signatures, hashing and signing, Hashed RSA, El Gamal and DSA signature schemes, public-key infrastructures, certificates, cryptography in TLS, IPSec and virtual private networks, application case study 2: secure online purchasing

Lecture 11: Key protocols: key transport, key agreement, notions and schemes (e.g., Diffie-Hellman schemes); key management: concepts and lifecycle; code obfuscation, application case study 3: digital rights management; brief discussion of quantum computing and quantum-resistant cryptography

Lecture 12: Key lengths and recommendations, user authentication: password, challenge-response and zero-knowledge protocols; server authentication; application case study 4: secure online banking; secure 2-party and multi-party function evaluation, application case study 5: sugar beet auction; digital cash, cryptocurrencies, application case study 6: bitcoin, blockchain, zcash

Readings

The **required** texts for the course are:

1) [KL] J. Katz and Y. Lindell, Introduction to Modern Cryptography: Principles and Protocols, Chapman & Hall/CRC Press, 2nd edition (see http://www.cs.umd.edu/~jkatz/imc.html)
2) [MOV] A. Menezes, P. Van Oorschot, S. Vanstone, Handbook of Applied Cryptography, CRC Press, August 2001, fully available online at http://www.cacr.math.uwaterloo.ca/hac/)

Note: text (1) contains about 85% of the class material; text (2) contains about 50% of the class material; past cs6903 students typically found it easier to study on lecture slides and to use (1) to strengthen understanding.

Optional and **recommended** texts are:

1) One among the following two texts:
    a. [FSK] N. Ferguson, B. Schneier and T. Kohno, Cryptography Engineering: Design, Principles and Practical Applications, Wiley Publishing, Inc., 2010 (this book gives exposure to more cryptography engineering aspects and might be considered a modern follow-up of (1b), the first book that was written on the topic)
    b. B. Schneier, Applied Cryptography, 2nd edition, J. Wiley and Sons.
2) W. Stallings, Cryptography and Network Security: Principles and Practice, 2nd edition, Prentice Hall.
3) Books at http://www.freetechbooks.com/information-security-f52.html

Course requirements

Homework assignments are multiple-choice tests where you are presented a set of 15 questions and are required to choose the correct answer among 4 possible ones. You will also be asked to provide a brief rationale for your answer to these questions. For i=1,..,4, homework hw[i] typically refers to lectures 3i-2, 3i-1, 3i and is due by the end of the week dedicated to Lecture 3i. Time permitting, the instructor may go over hw problem explanations and hw solutions either during the weekly online meeting or through a recorded video made available online or both. Homeworks should be considered (the only) practice problems for your midterm and final exams. Each of your

answers will be automatically scored by the web application depending on whether it is correct (5 points per answer) or not (0 points); this score should only be considered as feedback. For gradebook and class grade purposes, each of your answers receives the max available score if your rationale just shows that you tried to answer it, regardless of whether your solution is correct or not. For more successful practice for midterm and final, the student should answer hw questions as if they were midterm or final questions, spending no more than 25 minutes on each question after having secured their understanding of the related lectures.

The **projects** consist of solving practical problems (via software implementation) including: (1) the breaking or design of (variations of) a number of known cryptographic primitives (e.g., encryption schemes, authentication schemes, etc.); and (2) designing and implementing privacy and security solutions, based on the cryptography studied in class, as an improvement to a real-life system. In addition to implementation tasks, the projects will require a presentation document possibly including details on software documentation, cryptography design, cryptanalysis, performance analysis, etc. A project will have to be realized by a team of a number of students (to be decided during the class), and comes with a minimal assignment. Any additional work performed by the student(s) will be considered extra credit work. We will have a workshop day where each student will have a chance to present at least one of the projects. Students with the best projects will receive an increase on their class grade. Authorized sources: personal notes, required or recommended textbooks, and other web sources (only if they are properly quoted and not plagiarized). No collaboration is authorized with students beyond your team, or with parties external to the course. The following collaboration with instructor and TAs is authorized: you can ask via e-mail or during office hours for clarifications on the project problem statement (or about any solutions, after grades have already been posted).

The **midterm** is based on lectures 1-6; you will be given between 6 and 10 questions for which the answers may require writing a brief rationale (i.e., as for homework questions), and you will be given a time limit (i.e., 3 hours) to write your answers. The midterm is a proctored test where you are allowed to inspect your textbook and lecture slides. For more detailed info on authorized (or not) sources, see info posted by the instructor. If the instructor is proctoring your exam (which most likely happens to onsite students but not to online students), you can ask the instructor for clarifications on question statements. No collaboration is authorized with anyone else.

The **final** is based on lectures 7-12; you will be given between 6 and 10 questions for which the answers may require writing a brief rationale (i.e., as for homework questions), and you will be given a time limit (i.e., 3 hours) to write your answers. The final is a proctored test where you are allowed to inspect your textbook and lecture slides. For more detailed info on authorized (or not) sources, see info posted by the instructor. If the instructor is proctoring your exam (which most likely happens to onsite students but not to online students), you can ask the instructor for clarifications on question statements. No collaboration is authorized with anyone else.

The following opportunities for **extra credit** (for which collaboration among students is allowed) will be offered:

1) seminar attendance,
2) useful feedback provided to the instructor at the end of the semester,
3) additional work on projects,
4) design/implementation problems,
5) advanced topic surveys,
6) lecture notes writing.

You can take opportunities 1,2 and no more than one among 3,4,5,6.

All your work (homework, projects, midterm, final, extra credit) should be submitted on NYU Classes, under Tests and Quizzes, or will likely be not evaluated for class grade purposes.

Generally, students are very much encouraged to **e-mail** their **technical and non-technical questions** (including class organization, etc.) **to the instructor**; questions can be sent at any time, and an answer would likely appear within 1 or 2 business days. For real-time answers, office hours are more appropriate. For questions to fellow students, the online **forum** is more appropriate. The instructor is also happy to make time for a limited number of one-on-one office hours outside of the designated times. The diligent student is expected to generate a weekly list of unclear technical issues to clarify with the instructor and/or the TA through the above channels.

The student's course grade will be tentatively determined as a weighted average, as follows: class participation (5%), homework (10%), project1 (15%), project2 (20%), midterm (25%), final (25%). Class participation will be measured by the amount of technically interesting questions, answers, and observations that the instructor receives from the student. Weights are tentative and might slightly change (for instance, slightly increasing the weight of the final is possible). Incomplete grades will not be given, unless under exceptional circumstances. Homework submitted later than the due date is eligible for partial credit that decreases with time. Submitting homework late will always be more convenient than submitting no homework but never more convenient than submitting homework before the due date. The instructor will not drop students' lowest homework score.

The course' schedule and **due student activities** are summarized below:**ct**

Week	Class Events	What's due?
1	Lecture 1, Meeting 1 (Tu, Sep 4)	
2	Lecture 2, Meeting 2 (Tu, Sep 11)	
3	Lecture 3, Meeting 3 (Tu, Sep 18)	**HW1 on Lectures 1-3 (Sun, Sep 23)**
4	Lecture 4, Meeting 4 (Wed, Sep 26)	
5	Lecture 5, Meeting 5 (Tu, Oct 2)	
6	Legislative day (no class on Tu, Oct 9)	**Project 1 (Sun, Oct 14)**
7	Lecture 6, Meeting 6 (Tu, Oct 16)	**HW2 on Lectures 4-6 (Fri, Oct 19)**
8	Midterm exam (Tu, Oct 23)	
9	Lecture 7, Meeting 7 (Tu, Oct 30)	
10	Lecture 8, Meeting 8 (Tu, Nov 6)	
11	Lecture 9, Meeting 9 (Tu, Nov 13)	**HW3 on Lectures 7-9 (Sun, Nov 18)**
12	Lecture 10, Meeting 10 (Tu, Nov 20)	
13	Lecture 11, Meeting 11 (Tu, Nov 27)	
14	Lecture 12, Meeting 12 (Tu, Dec 4)	**Project 2 (Sun, Dec 9)**
15	Workshop: project presentations and final review (Tu, Dec 11)	**HW4 on Lectures 10-12 (Thu, Dec 13)**
16	Final exam (Tu, Dec 18)	**Extra credit (Thu, Dec 20)**

Moses Center Statement of Disability

If you are student with a disability who is requesting accommodations, please contact New York University's Moses Center for Students with Disabilities at 212-998-4980 or mosescsd@nyu.edu. You must be registered with CSD to receive accommodations. Information about the Moses Center can be found at www.nyu.edu/csd. The Moses Center is located at 726 Broadway on the 3rd floor.

Together with the above authorized (or not) sources and collaboration, this course follows the NYU School of Engineering policies and procedures on academic misconduct, as described below. Please read and follow this policy very carefully. In addition to this policy, this course's

Department asks this and other classes that evidence of disallowed collaboration, plagiarism, or other forms of cheating is punished with <u>huge score or grade penalties;</u> specifically:

1. **at the first occurrence, you get a 0 score to the specific assignment and your name is entered on a department list;**
2. **at the second occurrence, regardless of whether the first occurrence happened on this class or a previous one, you get a course grade of F on this class.**

NYU School of Engineering Policies and Procedures on Academic Misconduct

A. Introduction: The School of Engineering encourages academic excellence in an environment that promotes honesty, integrity, and fairness, and students at the School of Engineering are expected to exhibit those qualities in their academic work. It is through the process of submitting their own work and receiving honest feedback on that work that students may progress academically. Any act of academic dishonesty is seen as an attack upon the School and will not be tolerated. Furthermore, those who breach the School's rules on academic integrity will be sanctioned under this Policy. Students are responsible for familiarizing themselves with the School's Policy on Academic Misconduct.

B. Definition: Academic dishonesty may include misrepresentation, deception, dishonesty, or any act of falsification committed by a student to influence a grade or other academic evaluation. Academic dishonesty also includes intentionally damaging the academic work of others or assisting other students in acts of dishonesty. Common examples of academically dishonest behavior include, but are not limited to, the following:

1. Cheating: intentionally using or attempting to use unauthorized notes, books, electronic media, or electronic communications in an exam; talking with fellow students or looking at another person's work during an exam; submitting work prepared in advance for an in-class examination; having someone take an exam for you or taking an exam for someone else; violating other rules governing the administration of examinations.
2. Fabrication: including but not limited to, falsifying experimental data and/or citations.
3. Plagiarism: intentionally or knowingly representing the words or ideas of another as one's own in any academic exercise; failure to attribute direct quotations, paraphrases, or borrowed facts or information.
4. Unauthorized collaboration: working together on work that was meant to be done individually.
5. Duplicating work: presenting for grading the same work for more than one project or in more than one class, unless express and prior permission has been received from the course instructor(s) or research adviser involved.
6. Forgery: altering any academic document, including, but not limited to, academic records, admissions materials, or medical excuses.