The University of North Carolina at Greensboro
CSC 580: Cryptography and Security in Computing
Prof. Stephen R. Tate

Handout 12
April 28, 2014

Review Sheet for Final Exam

The final exam in this class will be Friday, May 2, from 7:00 – 10:00 (it will be designed to take no more than 90 minutes, but you will have the entire three hours if you'd like it). The exam will focus primarily on material covered since the midterm exam, but will include one or possibly two questions from material covered on the midterm, as well as questions which try to pull everything together. Make sure you review the terminology that was introduced in these chapters (terms are given at the end of each chapter).

The midterm provided pretty thorough coverage of Chapters 1 – 6. The following is a list of the main topics we have covered since the last exam:

- Formal Security Models (Handout)
- Chapter 8 (Number theory for public key cryptography)
- Chapter 9 (Public key crypto basics and RSA)
- Chapter 10 (Diffie-Hellman, ElGamal, and Elliptic Curves)
- Chapter 11 (Cryptographic Hash Functions)
- Chapter 7 (Pseudo-random number generation — some from the book but also from NIST document, including the Dual_EC_DRBG method and flaws)
- Chapters 12&13 (MACs and Digital Signatures – basic ideas only)
- Readings (Case studies – TrueCrypt and Voting Machine Security)

The final exam will be similar to earlier exams, with the length being roughly one question more than the midterm. There will also be a "challenge" question that requires you to think more creatively about the topics we have covered. For this question you will have at least 2 alternative questions, and you only need to solve one of them. Here's a sample "challenge question":

- The country of Oceania has decided to outlaw encryption but to allow integrity-protecting technologies such as hash functions and digital signatures. Since the policy-makers know that hash functions are "one-way", they figure people can't use them for encryption. However, any secure hash function can be turned into a secure symmetric cipher. Devise a way to do this. Some answers are better than others, but I don't expect perfection here — you simply need to have a reasonable approach: describe why you think your particular technique is secure.

Below are some samples of the more standard exam questions.

- The "voting machine security" paper mentioned that this voting system used CBC mode in which the IV was always all 0's. Prove that this encryption scheme (CBC mode with a constant IV) is not IND-CPA secure. Your answer should be fully justified along the lines of the problems we did in the "formal security models" unit.

- The Miller-Rabin primality testing algorithm has two possible outputs — describe what they are and what they mean. How can you increase the reliability of the answer you get from the Miller-Rabin algorithm?

- Describe the Diffie-Hellman key exchange algorithm — either a diagram or formulas are acceptable, as long as the technique is clear. Show why this establishes a secret key, and describe why this is believed to be secure (state any computational assumption that is made in this justification).

- Joe makes a system using RSA for public key cryptography, but in the key generation routine uses the standard C library `rand()` function for random values, which uses a 15-bit value for a seed. Why is this insecure? Be very specific in your answer, describing a real, practical attack with an explanation of the time complexity of the attack.