University of Maryland
CMSC414 — Computer and Network Security
Professor Jonathan Katz

Review for Final

Note: This is a high-level summary of topics you should know for the final. It is *not* comprehensive, but is intended to highlight some basic and fundamental points. Unless stated otherwise, *everything covered in class or in an assigned reading is fair game.* (So please don't ask "is this on the exam?". Unless stated otherwise below, the answer is "yes".)

Lecture 1 I do want you to understand why security is hard, why security is always a trade-off, why there is more to security than computer security. You should understand the "Trusting Trust..." article, as well as the broader point it is making.

Lectures 2–8 (Cryptography)

- Why are definitions and proofs important? Kerchoffs' principle.
- Understand the difference between the private- and public-key settings. Know what are the differences between private-key encryption, message authentication, public-key encryption, and digital signatures. Please get the syntax right: e.g., signing is *not* the same as decryption; verification takes as input a message and a tag/signature; a block cipher is not an encryption schemes, etc. You should know names of schemes used in practice for each of these applications.
- Understand the security guarantees provided by various definitions. When (and why) must encryption be randomized? Does encryption also guarantee integrity? What is the difference between chosen-plaintext security and chosen-ciphertext security? How can you exploit usage of "weak" encryption in protocols?
- What is the one-time pad? What are its limitations? What does computational security mean? What is a block cipher?
- You should know about DES, AES; also ECB, CBC, and CTR modes and why they are insecure against chosen-ciphertext attacks. Hash functions; birthday attacks.
- Understand Diffie-Hellman, El Gamal, and RSA, and their weaknesses. (E.g., why are textbook RSA signatures/encryption insecure?) Hybrid encryption.
- Why does crypto not solve all security problems?
- You are not responsible for the papers from lecture 8. However, I may ask a question related to ATM/bank security especially as it pertains to HW2. You should also generally be aware of the silly crypto mistakes that tend to be made, so skimming the papers may be helpful.
- You are not responsible for any aspects of the JCA.

Lectures 9–12 (System security)

- Understand the distinction between policy and mechanism.
- Understand the Saltzer and Schroeder principles, as covered in class.
- Understand the distinction between authentication and authorization.
- Access control: MAC, DAC, RBAC. Different techniques and their trade-offs; different security policies.
- You do not need to read the papers from lecture 10; this material will not be on the exam. (You *do* need to know the differences between capabilities and ACLs.)
- Bell-LaPadula and Biba models, etc.
- Trusted computing.
- You do not need to know about memory protection (lecture 13, slides 2–12).

Lectures 13–21 (Network security)

- We spent a fair amount of time talking about key exchange and mutual authentication protocols, and all of this makes for good exam questions. Please make sure also to read the assigned sections of the book (as listed on the course syllabus).
- You do *not* need to read the papers assigned for lectures 17 or 18, but you are responsible for anything related to those papers that was covered in class.
- You do not need to know the zero-knowledge protocols I covered in lecture 19 (slides 19–22), or lecture 20 (slide 3). However, you should know what zero-knowledge is (at a high level) and what are some applications of it.

Lectures 22–24 (Database and PL security)

- You *should* read the assigned papers for lectures 22–24.
- Make sure to understand the advantages and disadvantages of different mechanisms for database privacy.
- I may give you a snippet of code and ask you to identify potential vulnerabilities, similar to (but easier than) what you did for HW4.
- You should be aware of some basic techniques for secure programming, and some ways of defending against buffer overflows.

Lectures 25–27 (Network security in practice)

- Firewalls and IDS will only be covered at a high level. You should, however, know Bayes' law and how to use it.
- You should understand the network stack model, and the different tradeoffs for implementing security at various layers of the stack.
- You do not need to memorize any details of IPSec, IKE, or SSL. You *should*, however, read the assigned sections of the book. I may present you with various pieces of these protocols and ask you to explain what they do, what would happen if some modification is introduced, or ask for a rationale for some decision.