YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

CPSC 467a: Cryptography and Computer Security *Handout #25*
Professor M. J. Fischer *December 13, 2005*

Study Guide for the Final Exam

1 Exam Coverage

The final exam will cover the entire course but with greater emphasis on the topics of lectures 12–24 that were not covered by the midterm exam. The course content is presented in several different formats:

1. In-person class lectures.
2. Written lecture notes, available on the course web site.
3. Written handouts, available on the course web site. I draw your attention to the four handouts on the more mathematical aspects of the course:
   - Handout 4: Number theory summary.
   - Handout 5: Linear congruence equations.
   - Handout 6: The Legendre and Jacobi symbols.
   - Handout 21: Pseudorandom sequence generation.
4. Textbook (Trappe and Washington), relevant sections from chapters 6–10, 12–14.
5. Other resources available in the library and on the web.
6. Problem sets and solutions.

2 Review Outline

Below I give a list of topics, concepts, definitions, theorems, algorithms, and protocols that we have covered since the midterm that I expect you to know. This list is not inclusive, as I'm sure I have missed some things. Please refer to the study guide for midterm examination for the topics covered in the first part of the course.

1. Number theory.
   (a) Primitive roots.
      - Lucas test.
      - Discrete logarithm.

   (b) Quadratic residues.
      - Square roots modulo a prime.
      - Square roots modulo a product of two distinct primes.
      - Euler criterion.
      - Finding square roots modulo prime p when $p \equiv 3 \pmod{4}$.

    - Legendre symbol.
    - Jacobi symbol.
    - Jacobi symbol identities (don't memorize, but understand what they are).
    - Computing the Jacobi symbol.
  - (c) Probabilistic primality testing
    - General framework for tests of compositeness (from lecture notes 11).
    - Fermat test of compositeness $\zeta_a(n)$.
    - Strassen-Solovay test of compositeness $\nu_a(n)$
    - Miller-Rabin test of compositeness $\mu_a(n)$.

2. Cryptographic protocols based on number theory (besides RSA).
   - (a) Diffie-Hellman key exchange.
   - (b) ElGamal key agreement.
   - (c) ElGamal public key cryptosystem.
   - (d) Goldwasser-Micali (QR) probabilistic cryptosystem.
3. Digital signatures.
   - (a) Definition of a digital signature.
     - Signature.
     - Authenticated (signed) message.
     - Signature function.
     - Verification predicate.
   - (b) RSA digital signature scheme.
     - Commutative cryptosystems.
     - Signatures from non-commutative cryptosystems.
   - (c) Security of digital signatures.
     - Kinds of forgery attacks.
     - Signing random messages.
     - Signing message digests.
     - Differences between conventional and digital signatures.
     - Disavowal.
4. Message digest (hash) functions.
   - (a) Desired properties.
     - One-way property.
     - Weak collision-free property.
     - Strong collision-free property.
   - (b) Relations among properties.
5. Combining signatures with encryption.
6. ElGamal signature algorithm.

7. Digital signature algorithm (DSA).

8. Common hash functions.

   (a) SHA–1.

   (b) MD5.

9. Extending fixed-length hash functions.

   (a) Doubling the reduction amount.

   (b) A general chaining method.

   (c) Hash functions do not always look random.

10. Birthday Attack.

    (a) Birthday paradox.

    (b) Relevance to hash functions.

11. Hash from cryptosystem.

    (a) Techniques for building hash from cryptosystem.

    (b) Possible problems with this approach.

12. Authentication using passwords.

    (a) Passwords.
      - Insecure when sent in clear.
      - Bob learns Alice's password during normal use.

    (b) Secure password storage.
      - Problems securing storage.
      - Encrypted passwords.

    (c) Dictionary attacks.
      - How does it work?
      - Salt.
      - Why does salt help?

13. Authentication while preventing impersonation.

    (a) Challenge-response authentication protocols.
      - Basic challenge-response idea.
      - Security problem having Alice sign random strings.
      - Attempts to patch basic protocol.

    (b) Simplified Feige-Fiat-Shamir authentication protocol.
      - Relies on difficulty of finding square roots mod $n = pq$.
      - Alice's secret is square root of public number.
      - Simple protocol must be repeated.

    (c) Cheating Alice.

- Must fool Bob on each of t rounds.
- Ability to answer both challenges on some round implies can compute $\sqrt{v^{-1}}$.
- Otherwise, success probability $\leq 1/2^t$.

(d) Cheating Bob.

- Restricted query set limits cheating Bob's power.
- Cheating Bob only learns random quadratic residue and one square root.

14. Interactive proofs.

(a) Secret cave protocol.

(b) ZKIP for graph isomorphism.

- Graph isomorphism.
- Isomorphism problem.
- Prover (Alice).
- Verifier (Bob).
- Why doesn't Bob learn Alice's s secret?

(c) Relationship between ZKIP and secret splitting.

(d) Interactive proof of graph non-isomorphism.

- Non-isomorphism problem is essentially different from isomorphism problem.
- Form of protocol is quite different.
- Probably not zero knowledge.

15. Bit commitment.

(a) Definition.

i. Commitment hides Alice's bit.
ii. Alice can open her commitment.
iii. Alice can't reveal incorrect value.

(b) Use of bit-commitment ideas in non-isomorphism protocol.

16. Definition of zero knowledge via simulation.

(a) Zero-knowledge is property of Alice's protocol only.

(b) Compares power of arbitrary algorithms:

- That interact with Alice.
- That do not interact with Alice.

(c) Alice's protocol is zero knowledge if both kinds of algorithms have same power.

17. Composing Zero-Knowledge Proofs.

(a) Serial execution.

- Preserves zero knowledge.
- Many interactions.

(b) Parallel execution.

- Not provably zero knowledge.

- More efficient, fewer interactions.

18. Full Feige-Fiat-Shamir authentication protocol.

    (a) Combines parallel and serial execution ideas.

    (b) Provably zero knowledge when parallelism sufficiently restricted.

    (c) Less bandwidth than parallel execution of simple protocol.

19. Non-interactive "interactive" proofs.

    (a) Alice simulates interactive proof; sends result to Bob.

    (b) Bob needs assurance that Alice can't "cook" challenge bits.

    (c) Allows Bob to impersonate Alice to Carol by replaying proof.

    (d) Feige-Fiat-Shamir signatures.

    - Based on non-interactive version of FFS protocol.
    - Uses hash function to generate query bits.

20. Secret splitting.

    (a) Splitting secret into two shares using XOR.

    (b) Splitting secret into multiple shares.

    - Definition of what a (τ, k) threshold scheme is.
    - Lagrange interpolation
    - Shamir's scheme based on polynomials.
    - Why fewer than τ shares give no information about secret.

    (c) Extensions.

    - Verifiable secret sharing.
    - Fault tolerance.

21. Bit-commitment problem.

    (a) Terminology and concepts.

    - Commitment, blob, cryptographic envelope.
    - $\mathbf{commit}(b)$.
    - $\mathbf{open}(c)$.

    (b) Commitment using symmetric cryptography.

    - Why one-time pad does not work for bit-commitment.
    - Commitment protocol using suitable symmetric cryptosystem.
    - Commitment protocol using hash functions.
    - Commitment protocol using PRSG.

    (c) Abstract bit-commitment schemes.

    - Defining properties.
    - Generic protocol.

22. Coin-flipping problem.

   (a) Problem definition.
   (b) Solution using blobs.

23. Locked box paradigm.
   (a) Coin-flipping using boxes with multiple locks.
   (b) Commutative cryptosystems.
      - Definition.
      - RSA variation useful for implementing electronic locked boxes.
      - Security properties of the RSA variation.
   (c) Coin-flipping protocol using commutative cryptosystems.
   (d) Card dealing using locked boxes.

24. Oblivious transfer.
   (a) Oblivious transfer of a secret.
      - The problem.
      - Rabin's protocol.
      - Potential problem with Rabin's protocol.
      - Fix using zero-knowledge proof of knowledge of a square root.
   (b) One-out-of-two oblivious transfer.
      - The problem.
      - Protocol using two PKS key pairs.

25. Pseudorandom sequence generation.
   (a) Concepts:
      - What is a PRSG?
      - Why does the seed need to be random?
      - What does it mean for a string to look random?
   (b) Blum-Blum-Shub PRSG
      - Blum integers.
      - BBS algorithm.

26. Notions of randomness.
   (a) Indistinguishability.
      - Probabilistic polynomial time Turing machine.
      - Judges.
      - What it means to be a cryptographically strong PRSG.
   (b) Next-bit prediction.
   (c) Building a judge from a next-bit predictor.
   (d) Previous-bit prediction.
   (e) Construction of a next-bit predictor from a previous-bit predictor.

(f) Security of BBS PRSG.

27. Kerberos.

    (a) Parties involved: Trusted server, Alice, Bob.

    (b) Uses symmetric cryptography only.

    (c) Simplified protocol (see notes).

    - Nonce.
    - Timestamp.
    - Lifetime.
    - Ticket.
    - Authenticator.
    - Why is authenticator needed? Why isn't a ticket enough?

    (d) Security of Kerberos.

    - What kinds of attacks was Kerberos designed to resist?
    - What kinds of attacks is it possibly subject to?

    (e) Practical considerations in real protocol.

28. Secure shell (SSH)

    (a) Purpose: secure communication between terminal and server.

    (b) Asymmetric cryptography used for authentication and key exchange.

    (c) Traffic encrypted with randomly-generated session key.

    (d) Designed for ease of use.

29. Secure socket layer (SSL)

    (a) Encrypts traffic at the ISO transport layer.

    (b) Basis for Transport Layer Security (TLS) protocol (compatible with SSLv3).

    (c) Operates between peers.

    (d) Uses X.503(v3) certificates to obtain public keys.

    (e) Performs secure key exchange to establish a session key.