YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

CPSC 467b: Cryptography and Computer Security *Handout #7*
Professor M. J. Fischer *February 20, 2012*

Study Guide to Mideterm Exam

For the exam, you are responsible for the contents of lectures 1–12, the concepts used in problem sets 1–3, and the corresponding sections of the Trappe and Washington textbook. In greater detail, you are responsible for the following chapters and sections of **Trappe and Washington**:

- Chapter 1 [Overview].
- Chapter 2 [Classical Cryptosystems], sections 2.1–2.4, 2.6 (Playfair Cipher only), 2.7–2.9.
- Chapter 3 [Number Theory], sections 3.1–3.3, 3.5–3.7.
- Chapter 4 [DES], sections 4.1–4.6.
- Chapter 5 [AES].
- Chapter 6 [RSA].
- Chapter 7 [Discrete Log], sections 7.1, 7.2, 7.4, 7.5.
- Chapter 8 [Hash Functions], section 8.1–8.3.
- Chapter 9 [Digital Signatures], sections 9.1–9.3, 9.5.
- Chapter 11 [Digital Cash].

Goldwasser and Bellare is a supplement to the lectures and the main textbook. It contains material that goes way beyond what this course is able to cover, and you are not responsible for material in it that was not covered in the lectures. Nevertheless, you might find the following sections helpful in understanding some of the class material:

- Chapter 1 [Modern Cryptography].
- Chapter 2 [One-way and Trapdoor Functions], sections 2.1, 2.2.1, 2.3.1, 2.3.2.
- Chapter 4 [DES and AES], sections 4.2 and 4.5.
- Chapter 6 [Private-key Encryption], sections 6.1, 6.2.
- Chapter 7 [Public-key Encryption], sections 7.1, 7.2.3, 7.2.4, 7.3.
- Chapter 8 [Hash Functions], section 8.1.
- Chapter 9 [Message Authentication], sections 9.1, 9.2, 9.3, 9.7.1.
- Chapter 10 [Digital Signatures], sections 10.1, 10.2, 10.3.1–3.

Below is an **index to the lecture notes**. It lists all of the sections, subsections, and slide titles from lectures 1–12.

1 Course Overview [lecture 01]

- What is this course about?
- Role of cryptography
- Information security in the real world
- How is security achieved in the real world?
- Threat examples
- Principles of risk management
- Focus of this course
- Computer science, mathematics and cryptography
- Organization of this course
- What this course is not
- Example primitive: Symmetric cryptography (informal)
- Application of a symmetric cryptosystem
- Solution using symmetric cryptography
- Eve's side of the story
- Requirements

2 Symmetric Cryptography [lecture 01]

- Symmetric cryptosystems (somewhat more formal)
- Desired properties
- What's wrong with this definition?

3 Security of Symmetric Cryptography [lecture 02]

- Choosing a cryptosystem
- An analogy—choosing a car
- Quantifying computational difficulty
- Some important questions
- Modern Cryptography
- Giving precise answers to security questions

3.1 Complexity [lecture 02]

- Measuring computational difficulty
- Role of complexity theory
- Feasibility

3.2 Confidentiality [lecture 02]

- Keeping data confidential
- A more nuanced approach
- Attacks that do not always succeed

3.3 Attacks [lecture 02]

- Eve's information
- Attack scenarios
- Known plaintext attacks
- Chosen text attack scenarios
- Why would Alice cooperate in a chosen plaintext attack?
- Adaptive chosen text attack scenarios

3.4 Randomness [lecture 02]

- Randomness in cryptography
- Independence
- Joint probability distribution
- Eve's success probability
- Computational security
- Practical security considerations

4 Probability Theory [lecture 02]

- Probability distributions and events
- Random variables
- Experiments
- Conditional probability
- Statistical independence

5 Perfect Secrecy [lecture 02]

- Information-theoretic security
- Base Caesar cipher
- Full Caesar cipher

6 Perfect secrecy [lecture 03]

6.1 Caesar cipher [lecture 03]

- Caesar cipher
- Simplified Caesar cipher
- Joint message-key distribution
- Conditional probability distribution
- Proof of perfect secrecy

6.2 Loss of perfection [lecture 03]

- A minor change
- Perfect secrecy lost
- Caveats with perfect secrecy
- Known plaintext attack against simplified Caesar cipher

- Man-in-the-middle attacks
- Modification attack against base Caesar cipher
- A modification attack on English vowels
- A general's orders
- Moral

7 Classical ciphers [lecture 03]

7.1 One-time pad [lecture 03]

- One-time pad
- The one-time pad cryptosystem
- One-time pad in practice
- One-time pad vulnerable to a known plaintext attack
  f03

7.2 Affine ciphers [lecture 03]

- Affine ciphers

7.3 Polyalphabetic ciphers [lecture 03]

- Polyalphabetic ciphers
- Vigenère cipher

7.4 Hill cipher [lecture 03]

- Hill cipher

7.5 Playfair cipher [lecture 03]

- Playfair cipher
- How Playfair works
- Example Playfair matrix
- Encrypting in Playfair: preparing the message
- Encrypting in Playfair: substituting the pairs
- Decrypting in Playfair

8 Block ciphers [lecture 03]

- Block ciphers
- Analysis of the Caesar cipher
- References

9 Cryptanalysis [lecture 04]

- Breaking the Caesar cipher: An example
- Breaking the Caesar cipher: Extending these ideas
- Breaking the Caesar cipher: Conclusion

9.1 Brute force attack [lecture 04]

- Trying all keys
- Automating brute force attacks
- Random English-like messages
- Determining likely keys
- How long should the keys be?
- What is safe today and into the future?

9.2 Manual attacks [lecture 04]

- Cryptography before computers
- Monoalphabetic ciphers
- How to break monoalphabetic ciphers

10 Building block ciphers [lecture 04]

10.1 Building blocks [lecture 04]

- Substitution: Replacing one letter by another
- Transposition: Rearranging letters
- Composition: Building new ciphers from old
- Practical ciphers

11 Data Encryption Standard (DES) [lecture 04]

- Data encryption standard (DES)
- Feistel networks
- DES Feistel network
- One stage
- Properties of Feistel networks
- Obtaining the subkey
- The scrambling function
- DES scrambling network
- Connecting the boxes
- Expansion permutation
- Connecting the outputs
- Security considerations

12 Using block ciphers [lecture 04]

- Block ciphers

12.1 Padding [lecture 04]

- Using a block cipher
- Padding
- Padding rules
- Padding example

13 References [lecture 04]

14 Advanced Encryption Standard [lecture 05]

- New Standard
- Details
- More details
- How does AES actually work?
- Confusion & Diffusion
- Transformations
- Roles of the four transformations
- Roles of the four transformations
- Preliminaries
- SubBytes()
- SubBytes() S-Box
- SubBytes()
- ShiftRows()
- MixColumns()
- Matrix multiplication
- AddRoundKey()
- Decryption
- AES Security
- Bruce Schneier on AES security

15 AES Alternatives [lecture 05]

- Other ciphers
- IDEA (International Data Encryption Algorithm)
- Blowfish
- RC6
- TEA (Tiny Encryption Algorithm)
- Additional Resources

16 Byte padding [lecture 06]

- Padding revisited
- PKCS7 padding

17 Chaining modes [lecture 06]

- Chaining mode
- Electronic Codebook Mode (ECB)
- Cipher Block Chaining Mode (CBC)
- Output Feedback Mode (OFB)
- Cipher-Feedback Mode (CFB)
- OFB, CFB, and stream ciphers
- Propagating Cipher-Block Chaining Mode (PCBC)
- Recovery from data corruption
- Other modes

18 Stream ciphers [lecture 06]

18.1 Symmetric cryptosystem families [lecture 06]

- Symmetric cryptosystem families

18.2 Stream ciphers based on keystream generators [lecture 06]

- Structure of stream cipher
- Key stream generator
- Security requirements for key stream generator
- Cryptographically strong pseudorandom sequence generators
- Ideas for improving stream ciphers

18.3 Stream ciphers based on block ciphers [lecture 06]

- Building key stream generators from block ciphers
- Stream ciphers from OFB and CFB block ciphers
- Extended OFB and CFB modes
- Some notation
- Extended OFB and CFB similarities
- Shift register rules
- Comparison of extended OFB and CFB modes
- Downside of extended OFB
- Possible solution

18.4 Rotor machines [lecture 06]

- Rotor machines
- How a rotor machine works
- Key stream generation
- Key stream generation (cont.)
- Changing the permutation
- History

19 Steganography [lecture 06]

- Steganography

20 Active adversaries [lecture 06]

- Active adversary
- Some active attacks
- Replay attacks
- Fake encrypted messages
- Message-altering attacks
- Encrypting random-looking strings

21 Public-key cryptography [lecture 07]

- Public-key cryptography
- Asymmetric cryptosystems
- Security requirement
- Man-in-the-middle attack against 2-key cryptosystem
- Passive attacks against a 2-key cryptosystem
- Facts about asymmetric cryptosystems
- Hybrid cryptosystems

22 RSA [lecture 07]

- Overview of RSA
- RSA spaces
- Encoding bit strings by integers
- RSA key generation
- RSA encryption and decryption
- RSA questions
- Tools needed to answer RSA questions

23 Factoring Assumption [lecture 07]

- Factoring assumption
- How big is big enough?

24 Computing with Big Numbers [lecture 07]

- Algorithms for arithmetic on big numbers
- Big number libraries
- GMP
- Openssl crypto package

25 Fast Exponentiation Algorithms [lecture 07]

- Modular exponentiation
- Difficulty of modular exponentiation
- Controlling the size of intermediate results
- Efficient exponentiation
- Combining the m_i for general e
- Towards greater efficiency
- A recursive exponentiation algorithm
- An iterative exponentiation algorithm
- Correctness
- A minor optimization

26 Number theory [lecture 07]

- Number theory overview

26.1 Division [lecture 07]

- Quotient and remainder
- mod for negative numbers
- Divides

26.2 Modular Arithmetic [lecture 07]

- The mod relation
- Mod is an equivalence relation

he two-place07 • Canonical names

27 Number Theory Needed for RSA [lecture 08]

- Number theory needed for RSA
- How these facts apply to RSA

28 Z_n: The integers mod n [lecture 08]

28.1 Modular arithmetic [lecture 08]

- The mod relation
- Mod is an equivalence relation

he two-place08 • Canonical names

- Mod is a congruence relation

28.2 GCD [lecture 08]

- Greatest common divisor
- Computing the GCD
- Euclidean algorithm
- Euclidean identities
- Computing GCD without factoring
- Repeated subtraction
- Using division in place of repeated subtractions
- Full Euclidean algorithm
- Complexity of GCD

28.3 Relatively prime numbers, $\mathbf{Z}_n^*$, and $\phi(n)$ [lecture 08]

- Relatively prime numbers
- Euler's totient function $\phi(n)$
- Example: $\phi(26)$
- A formula for $\phi(n)$

29 Computing in $\mathbf{Z}_n$ [lecture 08]

29.1 Modular multiplication [lecture 08]

- Multiplication modulo n
- Example: Multiplication in $\mathbf{Z}_{26}^*$

29.2 Modular inverses [lecture 08]

- Example: Inverses the elements in $\mathbf{Z}_{26}^*$.
- Finding modular inverses
- Diophantine equations
- Existence of solution

29.3 Extended Euclidean algorithm [lecture 08]

- Extended Euclidean algorithm
- Finding all solutions
- Example of extended Euclidean algorithm
- Computing the triples
- Extracting the solution

30 Generating RSA Encryption and Decryption Exponents [lecture 08]

- Recall RSA exponent requirement
- Sampling from $\mathbf{Z}_n^*$
- How large is large enough?
- A lower bound on $\phi(m)/m$
- Expected difficulty of choosing RSA exponent e

31 Euler's Theorem [lecture 09]

- Repeated multiplication in $\mathbf{Z}_n^*$
- Euler's and Fermat's theorem
- An important corollary
- Application to RSA
- Messages not in $\mathbf{Z}_n^*$
- Why Alice might want to avoid sending messages not in $\mathbf{Z}_n^*$
- Why a random message is likely to be in $\mathbf{Z}_n^*$
- RSA works anyway

32 Generating RSA Modulus [lecture 09]

32.1 Finding primes by guess and check [lecture 09]

- Recall RSA modulus
- Generating random primes of a given length

32.2 Density of primes [lecture 09]

- Expected number of trials to find a prime
- Prime number function
- Prime number theorem
- Likelihood of randomly finding a prime

33 Primality Tests [lecture 09]

- Algorithms for testing primality
- Tests for primality
- Probabilistic primality testing algorithm
- Trading off non-termination against possibility of failure

33.1 Strong primality tests [lecture 09]

- Strong primality tests

33.2 Weak tests of compositeness [lecture 09]

- Weak tests
- Use of a weak test of compositeness
- Algorithm P_3 using a weak test
- Meaning of output **?**
- Finding a random prime
- Success probability for GenPrime(k)

33.3 Reformulation of weak tests of compositeness [lecture 09]

- Boolean test of compositeness
- Meaning of a Boolean test of compositeness
- Useful tests
- Sample use of a useful test
- Application to RSA

33.4 Examples of weak tests [lecture 09]

- Finding weak tests of compositeness
- The division test $\delta_a(n)$
- The Fermat test $\zeta_a(n)$
- Carmichael numbers (Fermat pseudoprimes)

34 RSA Security [lecture 09]

- Attacks on RSA

34.1 Factoring n [lecture 09]

- RSA factoring problem

34.2 Computing $\phi(n)$ [lecture 09]

- $\phi(n)$ problem

34.3 Finding d directly [lecture 09]

- Decryption exponent problem
- Factoring n knowing e and d
- Square roots of $1 \pmod{n}$
- Finding a square root of $1 \pmod{n}$
- Using a non-trivial square root of unity to factor n
- Randomized factoring algorithm knowing d
- Notes on the algorithm
- Example

34.4 Finding plaintext [lecture 09]

- A ciphertext-only attack against RSA
- Hardness of ciphertext-only attack

35 One-way and Trapdoor Permutations [lecture 10]

- One-way permutations
- Negligible functions
- One-way functions

- Fine points
- Trapdoor functions
- A more intuitive formulation of trapdoor function
- The RSA function

36 Discrete Logarithm [lecture 10]

- Logarithms $\mathrm{mod} p$
- Discrete log problem

37 Diffie-Hellman Key Exchange [lecture 10]

- Key exchange problem
- D-H key exchange overview
- D-H key exchange protocol
- Security of DH key exchange

38 ElGamal Key Agreement [lecture 10]

- A variant of DH key exchange
- Comparison with first DH protocol
- ElGamal cryptosystem
- Combining key exchange with underlying cryptosystem
- A hybrid ElGamal cryptosystem
- Randomized encryption
- Remarks about randomized encryption

39 Primitive Roots [lecture 10]

- Using the ElGamal cryptosystem
- Primitive root
- Number of primitive roots
- Primitive root example
- Lucas test
- Problems with the Lucas test
- Special form primes
- Density of special form primes

40 Message Integrity and Authenticity [lecture 11]

- Protecting messages
- Protecting integrity and authenticity

40.1 Message authentication codes [lecture 11]

- Message authentication codes (MACs)
- Creating an authenticated message
- Verifying an authenticated message
- Cheating
- Computing a MAC
- Protecting both privacy and authenticity
- Another possible use of a MAC
- The problem
- Example of a flawed use of a MAC

40.2 Asymmetric digital signatures [lecture 11]

- Asymmetric digital signatures
- Asymmetric digital signatures
- Fundamental property of a signature scheme

40.3 Implications of Digital Signatures [lecture 11]

- What does a digital signature imply?
- Disavowal
- Practical usefulness of digital signatures

41 Digital Signature Algorithms [lecture 11]

41.1 RSA digital signatures [lecture 11]

- RSA digital signature scheme
- Commutative cryptosystems

41.2 Signatures from non-commutative cryptosystems [lecture 11]

- Signatures from non-commutative cryptosystems
- Interchanging public and private keys

41.3 ElGamal digital signature scheme [lecture 11]

- ElGamal signature scheme
- Why do ElGamal signatures work?

42 Security of Digital Signatures [lecture 11]

42.1 Desired security properties [lecture 11]

- Desired security properties of digital signatures

42.2 Random signed messages [lecture 11]

- Forging random RSA signed messages
- Importance of random signed messages

42.3 Adding redundancy [lecture 11]

- Adding redundancy
- Security of signatures with fixed redundancy
- Forging signatures with fixed redundancy

42.4 Signing message digests [lecture 11]

- Signing message digests
- Forging signed message digests
- Solving for s'
- Solving for m'
- Other attempts
- More advantages of signing message digests

42.5 Signed encrypted messages [lecture 11]

- Signed encrypted messages
- Weakness of encrypt-and-sign
- A forgery-resistant signature scheme with no privacy
- Encrypt or sign first?

43 Practical Signature Algorithms [lecture 12]

43.1 Digital signature algorithm (DSA) [lecture 12]

- Digital signature standard
- DSA key generation
- DSA signing and verification
- Why DSA works
- Double remaindering
- Example mod 29 mod 7

43.2 Common hash functions [lecture 12]

- Common hash function
- SHA-1
- SHA-1 broken

43.3 MD5 [lecture 12]

- MD5 overview
- MD5 padding
- MD5 chaining
- MD5 block function
- MD5 scrambling function
- Further remarks on MD5

44 Digital Signatures with Special Properties [lecture 12]

44.1 Blind signatures [lecture 12]

- Electronic voting
- Validating ballots
- Blind signatures
- RSA blind signatures
- Electronic cash
- Producing a digital coin

44.2 Group signatures [lecture 12]

- Group signatures
- Properties of group signatures
- Properties of group signatures (cont.)

44.3 Short signatures [lecture 12]

- Short signatures
- Applications of short signatures

44.4 Aggregate signatures [lecture 12]

- Aggregate signatures