MATC16 Cryptography and Coding Theory
Gábor Pete
University of Toronto Scarborough

What you should know for the midterm test

Principles and goals of cryptography:

Kerckhoff's principle. Shannon's confusion and diffusion. Possible attack situations (ciphertext only, chosen plaintext, etc.). Possible goals of attacker, and the corresponding tasks of cryptography (confidentiality, data integrity, authentication, non-repudiation). [Chapter 1, plus page 38 and `http://en.wikipedia.org/wiki/Confusion_and_diffusion` for diffusion & confusion.]

Classical cryptosystems:

Number theory basics: infinitely many primes exist, basics of modular arithmetic, extended Euclidean algorithm, solving $ax+by=d$, inverting numbers and matrices (mod n). [Sections 3.1-3 and 3.8.]

Shift and affine ciphers. Their ciphertext only and known plaintext attacks. Composition of two affine ciphers is again an affine cipher. [Sections 2.1-2.]

Substitution ciphers in general. [Section 2.4]

Vigenère cipher. Known plaintext attack. Ciphertext only: finding the key length, then frequency analysis. [Section 2.3.]

Hill cipher. Known plaintext attack. [Section 2.7.]

One-time pad. LFSR sequences. Known plaintext attack, finding the recursion. [Sections 2.9 and 11.]

Basics of Enigma. [Section 2.12, up to middle of page 53.]

The DES:

Feistel systems, simplified and real DES (without the exact expansion functions and S-boxes and permutations, of course), how decryption works in these DES versions. How the extra parity check bits in the real DES key ensure error detection. How confusion and diffusion are fulfilled in DES. [Sections 4.1-2 and 4.4.]

Double and Triple DES. Meet-in-the-middle attack. (I mentioned here that one can organize the two lists of length n and find a match between them in almost linear time ($n \log n$) instead of the naive approach that would give only n^2, and hence would ruin the attack completely.

If you want more details, see `http://en.wikipedia.org/wiki/Binary_search_tree`, but this won't be on the test.) Basic idea of password security, salt. [Sections 4.6-8.]

Modes of operation for block ciphers: ECB, CBC, CFB, OFB. [Section 4.5].

The AES (Rijndael) algorithm:

Algebra prerequisite: definition of a group and a field. How to check if a $\mathbb{Z}_p[x]$ polynomial is irreducible, how to construct a finite field by taking polynomials modulo an irreducible polynomial, how to do arithmetic in this field (including extended Euclidean algorithm for polynomials), what is $GF(p^k)$. [Section 3.11, except for 3.11.2 and 3.11.3.]

Shannon's Substitution/Permutation networks, how they ensure confusion and diffusion. [This is not in the book. See `http://en.wikipedia.org/wiki/Substitution-permutation_network`.]

The basic structure of the AES algorithm, construction of the S-box. [Sections 5.1-2, except for the exact form of the key schedule.]

The RSA algorithm:

Number theory prerequisites: Chinese remainder theorem. If $x^2 \equiv y^2 \pmod{n}$ with $x \not\equiv \pm y \pmod{n}$, then n is composite; in fact, $\gcd(x \pm y, n) \neq 1, n$. Modular exponentiation (repeated squaring). Euler's $\phi(n)$ function, Fermat's little and Euler's theorem. [Sections 3.4-6.]

The algorithm. For $n = pq$, knowing $\phi(n)$ is equivalent to knowing the prime factors p, q. How to achieve authentication and non-repudiation in RSA. [Sections 6.1 and 6.7.]

Knowing that some care is needed what p, q, d, e to choose, even though you won't be asked to reproduce the statements. [Section 6.2.]

The Fermat primality test. [First third of Section 6.3.]

Fermat's factorization trick, Pollard's $p - 1$ algorithm. [First page of Section 6.4]