Cryptographic concepts:  
- encryption  
- decryption  
  
Symmetric Cryptosystem  
- how do encryption and decryption work?  
- Caesar Cipher

Block ciphers:  
- How do block ciphers work?  
- example block ciphers  
- block cipher mode (ECB vs CBC): comparison of these two modes

Public Key Cryptography  
- how do encryption and decryption work?

RSA Cryptosystem  
- Mathematical background  
   - prime numbers  
   - Greatest Common Divisor  
   - Modular Arithmetic  
   - Modular inverse: x\*y = 1 mod n  
   - Extended Euclid's Algorithm  
- Setup: n, p, q, e, phi(n), d  
- Keys: public key (n, e) and private key (d)  
- Encryption: Plaintext M in Zn, C = M^e mod n  
- Decryption: M = C^d mod n  
- expectation: given n, p, q, and e, know how to   
    -- encrypt a given plaintext M,   
    -- given d, decrypt a given ciphertext C  
    -- compute d from e, or vice versa

Cryptographic Hash Functions  
- concept and definition  
- common example hash functions

Session Keys  
- Definition  
- Benefits

Forward search attack  
- definition  
- application to real world scenarios

Needham-Schroeder Key Exchange Protocol  
- How to prove that Alice is talking to Bob?  
- How to prove that Bob is talking to Alice?

Public-Key Key Exchange  
- The scheme: Alice sends { { ks } dA } eB to Bob  
- What are the assumptions for this scheme to work?  
- Man-in-the-Middle Attack against ths simple scheme { ks } eB

PKI  
- motivation for certificates  
- CA  
- Validation of certificates  
- Key Revocation  
- common products: X.509 and PGP

Digital signatures  
- Public Key Digital Signatures: m || { m } d\_Alice  
- Signature validation

n-stage Linear Feedback Shift Register for Stream Ciphers  
  - Expectation: can computer key sequence given the parameters

n-stage Non-Linear Feedback Shift Register for Stream Ciphers  
  - Expectation: can computer key sequence given the parameters  
  
IPsec  
- Basic knowledge: the two IPsec modes, services provided (confidentiality, integrity)

Authentication: the determination of identity or role that someone has with respect to the resources that the OS controls.   
  - something that the person has (e.g., Barcodes, magnetic stripe cards, a smart card, a SIM card, RFID tag, or a radio token with secret keys)  
  - something that the person knows (e.g., password): brutal force attacks against password  
  - something that the person is (e.g., fingerprint): candidates, advantage, disadvantage  
       -- Requirements for Biometric Identification

Password  
- Dictionary Attacks: how to calculate the probability of guessing a password in specified period of time  
- Password salting: what is it? why is it useful?

Access Control List  
- Linux ACL  
- 10-character notation  
- Deny be default; path-based access control principle; DAC  
- Extended ACL: why is it useful?

- Windows permissions  
- Explicit deny permission  
- standard vs advanced permissions  
- NOT path-based access control  
- ACE precedence (book p143)

Capabilities  
- definitions and their difference

Conceptually, a capability is like the row of an access control matrix. Each subject

has associated with it a set of pairs, with each pair containing an object and a set of

rights. The subject associated with this list can access the named object in any of the

ways indicated by the named rights. Capabilities encapsulate object identity. When a process presents a capability on behalf of a user, the operating system examines the capability to determine both

the object and the access to which the process is entitled. This reflects how capabiliies

for memory management work; the location of the object in memory is encapsulated

in the capability. Without a capability, the process cannot name the object in a

way that will give it the desired access

Tagged architecture

– Bits protect individual words

• B5700: tag was 3 bits and indicated how word was to be treated

(pointer, type, descriptor, etc.)

• Paging/segmentation protections

– Like tags, but put capabilities in a read-only segment or page

• CAP system did this

– Programs must refer to them by pointers

• Otherwise, program could use a copy of the capability—which it

could modify

• Cryptography

– Associate with each capability a cryptographic checksum enciphered

using a key known to OS

– When process presents capability, OS validates checksum  
- what is capability amplification and why is it useful?

Amplification is the increasing of privileges. The idea of modular programming,

and especially of abstract data types, requires that the rights a process has over

an object be amplified.

Needed for modular programming

– Module passwordDB

module passwordDB ... endmodule.

– It saves all passwords in a file (e.g., /etc/passwd)

– It has method change\_password, uses can call it

– Only passwordDB module can read and alter /etc/passwd

• So normal user process doesn’t get capability, but needs it when

calling change\_password

– Solution: give process the required capabilities while it is running

in the module

Ring-Based Access Control  
- Procedure executing ring: r  
- How to use access bracket (a1, a2) to determine if data can be accessed  
- How to use access bracket (a1, a2) and call bracket (a2, a3) to determine whether procedure can be called

Propagated Access Control List (PACL)  
- How do read / write operations affect PACLs of subjects and objects?  
- How is PACL used to make access control decisions?  
- An object can have PACLs associated with more than one creators. In that case, only subjects common to all PACLs can access the object