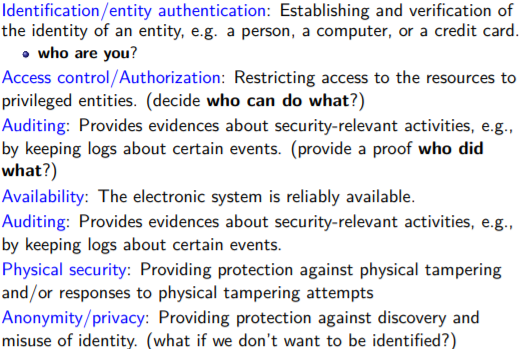
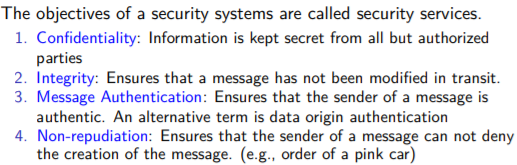
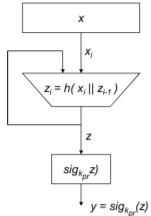
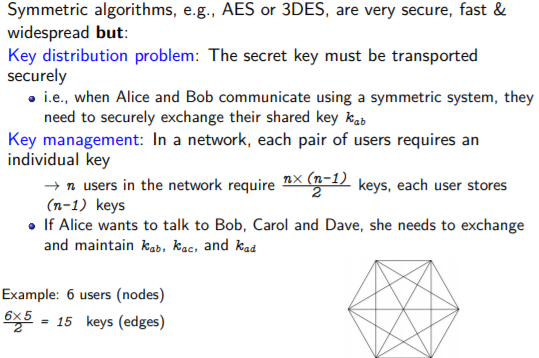
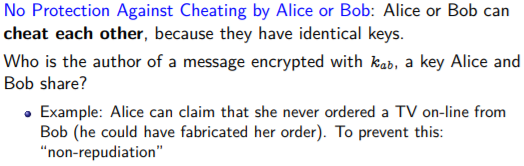
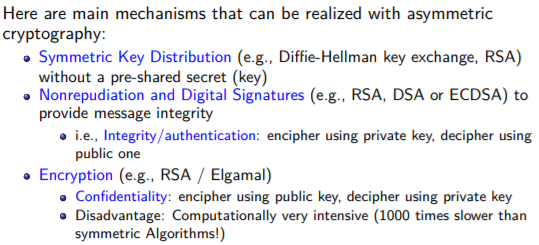
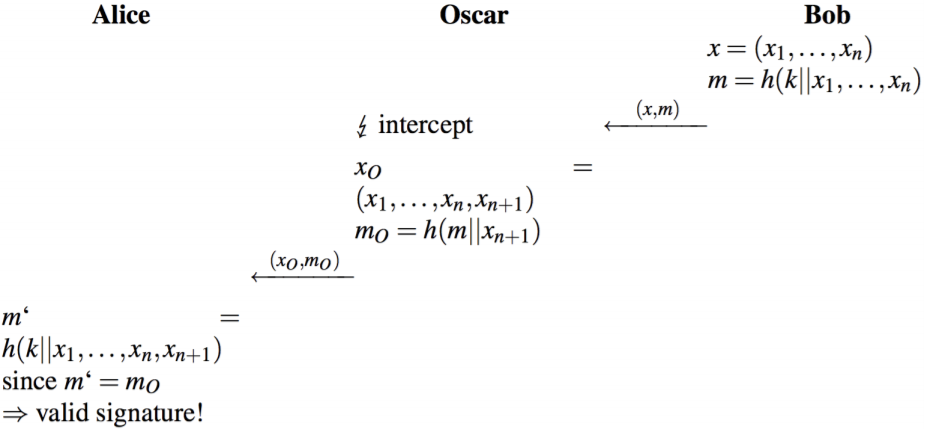
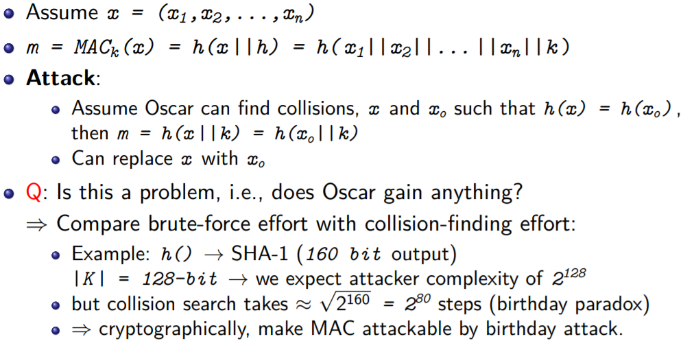
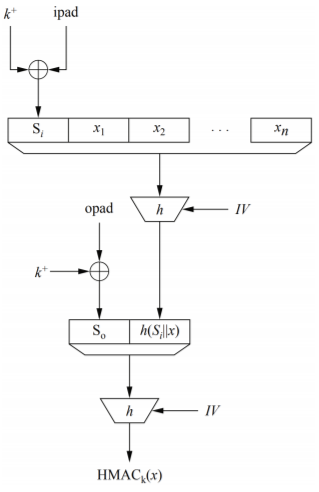
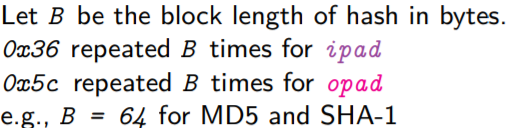
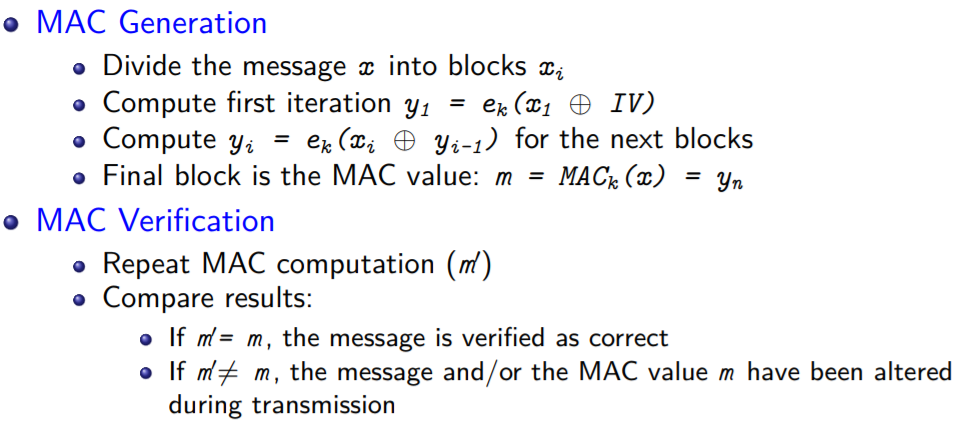
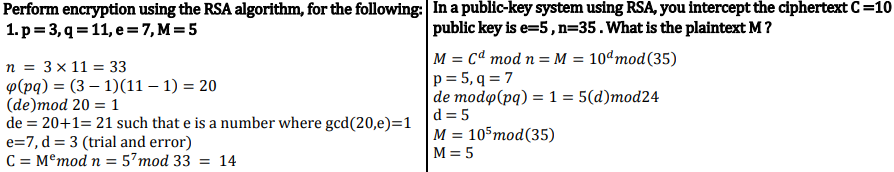
* Digital signatures
  + Motivation and principle of digital signatures
    - For a given message x , a digital signature is appended to the message
    - The signature is realized as a function with the message x and the private key as input.
    - The public key and the message x are the inputs to the verification function (hash).
  + Security services



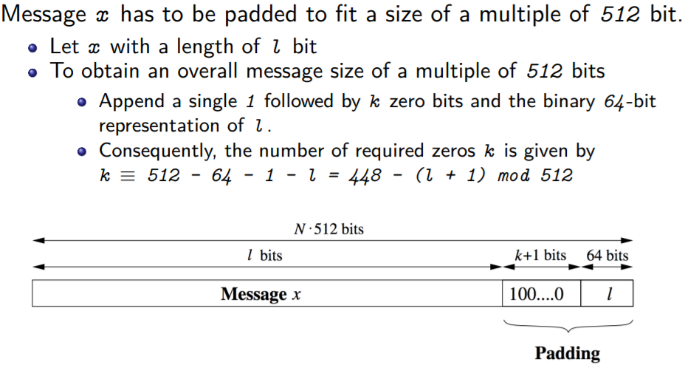
* + Digital signature schemes and attacks
    - Digital Signature Standard (DSA)
    - Elliptic Curve DSA
* Hash Functions
  + Motivation, security properties, attacks and algorithms
    - Naïve signing of long messages generates a signature of same length
    - Problems: Computational/double message overhead/security
    - Sign long messages with hash(x)
      * x has fixed length
      * z, y have fixed length
      * z, x do not have equal length in general
      * h(x) does not require a key.
      * h(x) is public.
    - Properties:
      * Compression
      * Efficiency
      * Preimage resist.: for any msg=z impossible to find any input x such that h(x)=z , i.e., h(x) is one-way
      * 2nd preimage resist: computationally infeasible to find any x’ s.t. h(x)= h(x’)
      * Collision Resist (**COLLISION ATTACK INFEASIBLE**)
* Symmetric and asymmetric encryption schemes
  + Using RSA, El-Gamal, ... to encrypt/decrypt
    - ELGAMEL
      * BAD: ciphertext is twice as long as the plaintext.
      * GOOD:the same plaintext gives a different ciphertext each time it is encrypted.
  + Symmetric Cryptography: Shortcomings
  +  
  + Practical Aspects, security mechanisms, and important Public-Key Algorithms
  + 

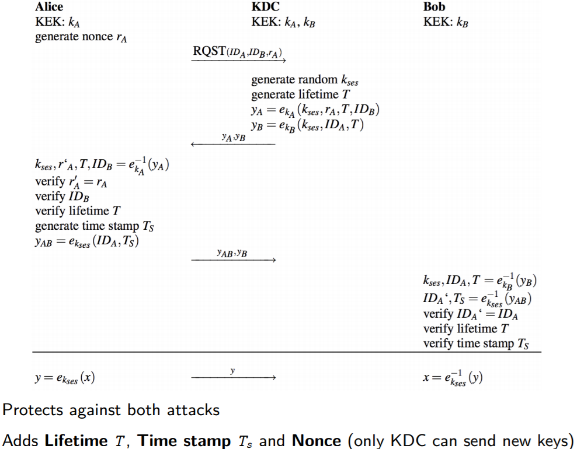
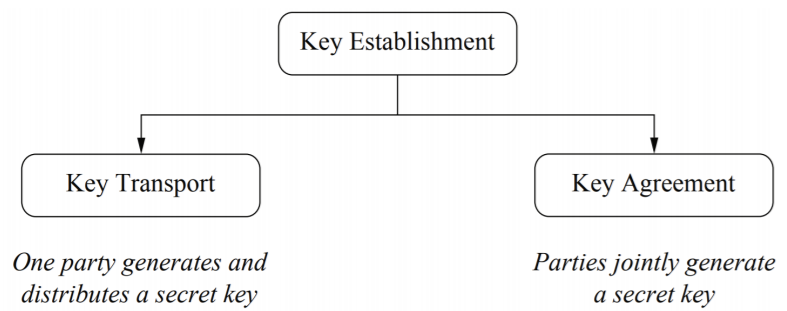
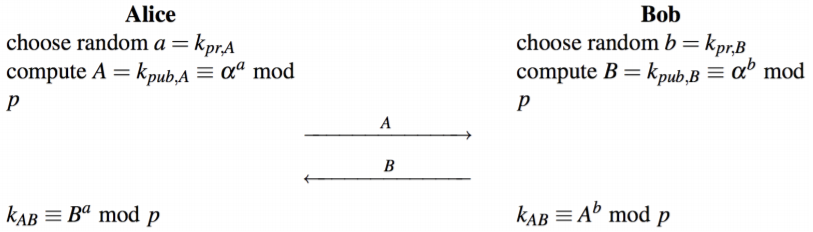
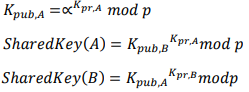
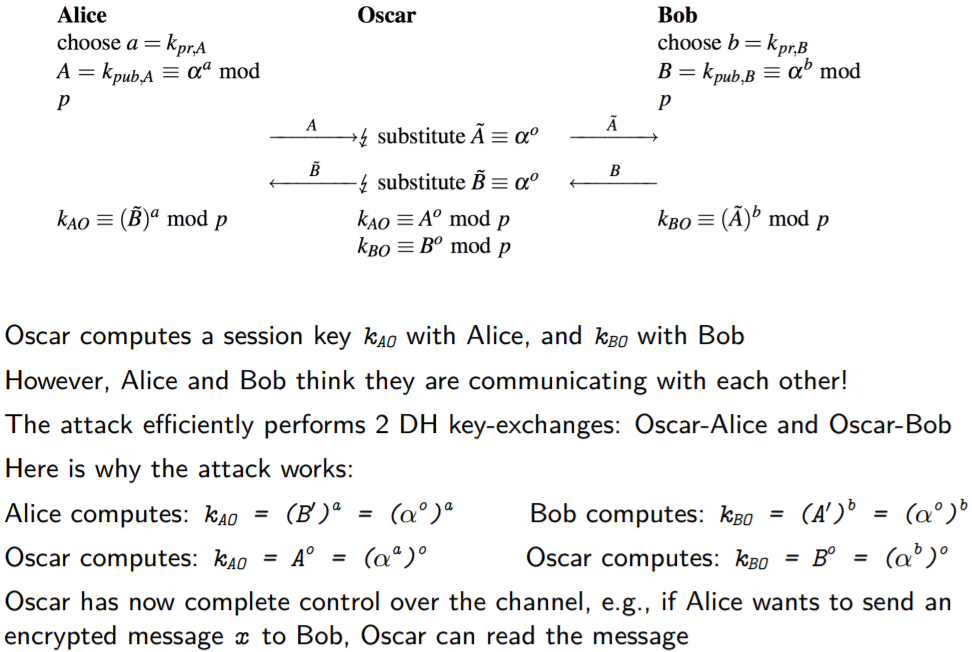
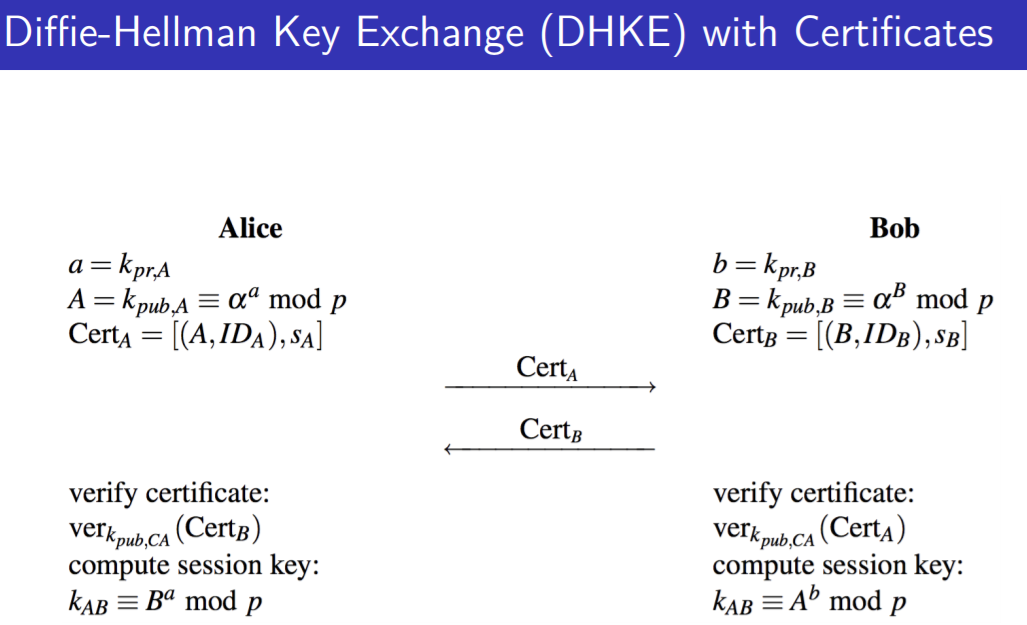
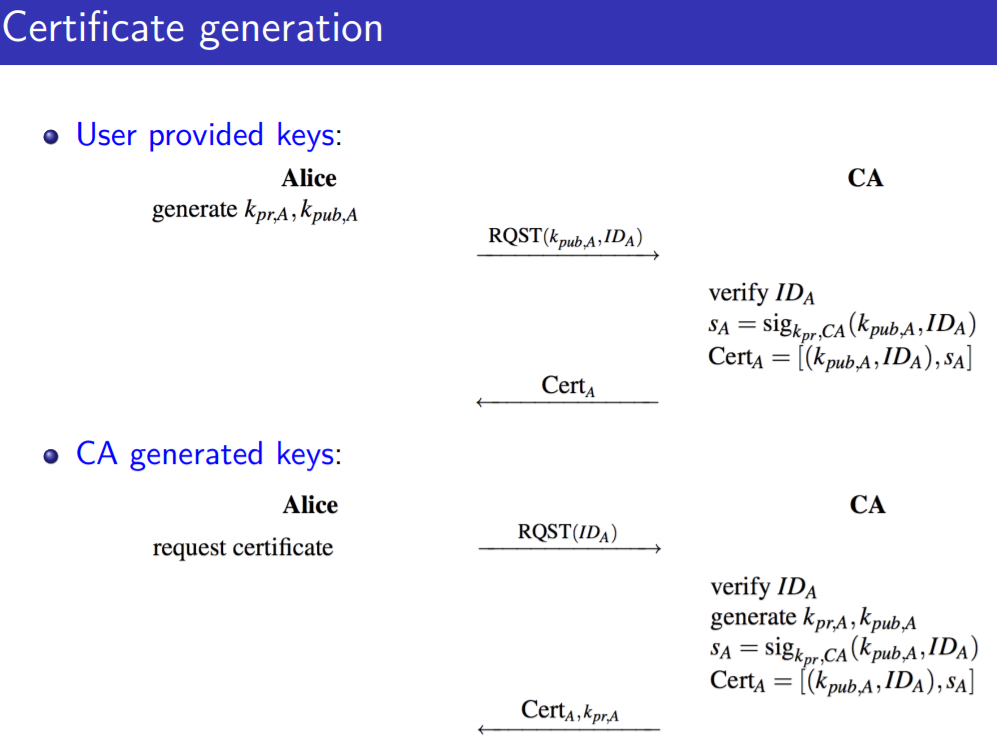
Message Authentication Codes (crypto checksums)

* + Motivation and principle behind MACs
    - Message auth.:: Bob computes m = MACk(x) and sends (x,m) to Alice. Alice receives (x,m’) and verifies that m’= m .
  + Security properties and attacks
    - 1. Cryptographic checksum: A MAC generates a cryptographically secure authentication tag for a given message.
    - 2. Symmetric: MACs are based on secret symmetric keys. The signing and verifying parties must share a secret key.
    - 3. Arbitrary message size: MACs accept messages of arbitrary length.
    - 4. Fixed output length: MACs generate fixed-size authentication tags.
    - 5. Message integrity: MACs provide message integrity: Any manipulations of a message during transit will be detected by the receiver.
    - 6. Message authentication: The receiving party is assured of the origin of the message.
    - 7. No nonrepudiation: Since MACs are based on symmetric principles, they do not provide nonrepudiation
    - 
    - Sec.suffix attackable by b-day attack:
  + MAC with hash functions and with block ciphers
    - HMAC in SSL/TLS
    - 
    - Chained Block Cipher MACs
    - 
    - Signatures verified by public key of asymmetric key pair
    - MACs verified by recipient via hash with shared secret key
  + RSA Encryption

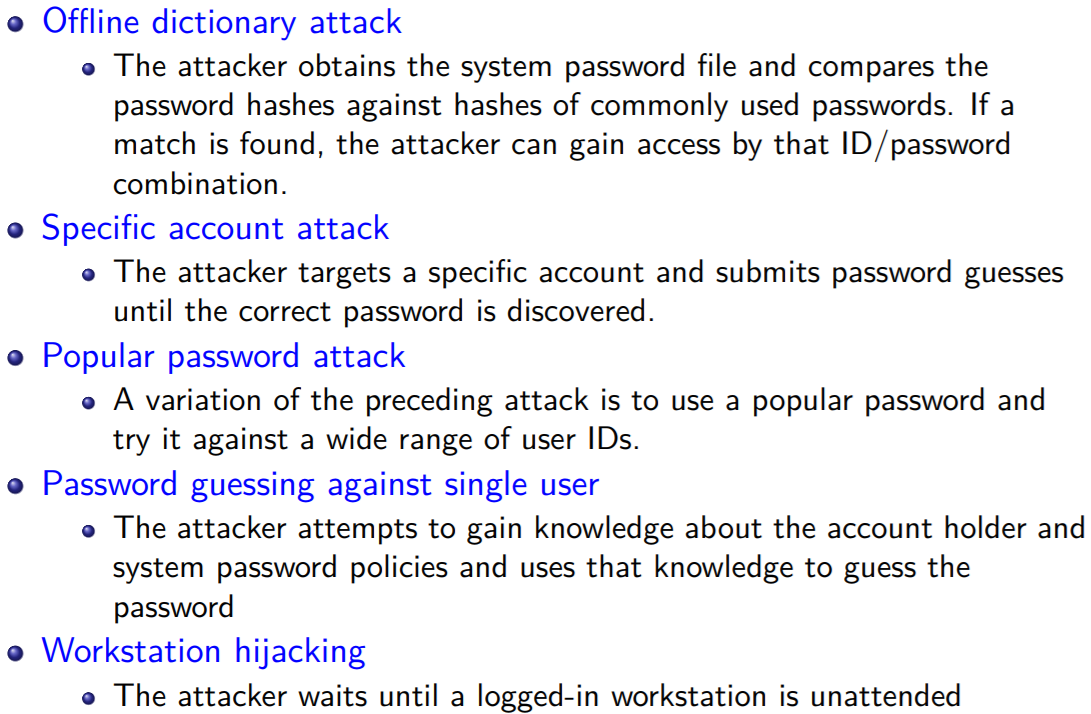
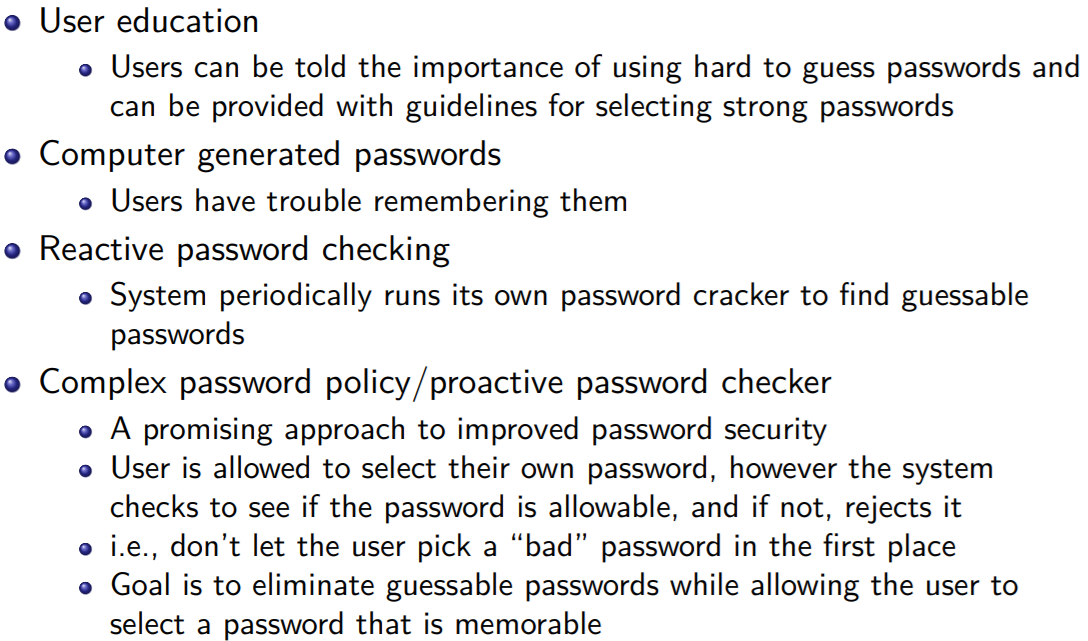
**RSA IS A PROBLEM OF FACTORING**

**DHKE IS A PROBLEM OF COMPUTATIONAL REDUCTION BY LOSS IN SECURITY (See formula diffs)**

**SHA-1:**

* Key Management & Identity
  + Key Distribution Problem
    - Each pair of users has a key (n-1 keys per user so n(n-1)~n^2 keys)
  + Classification of Key Establishment Methods.
    - 
  + Key Establishment with Key Distribution Center, Kerberos, …
    - Kerberos protects against: Replay atk/key confirmation atk (intercept)
    - **Client create authenticator**
    - **Authenticator creates “tickets” for KDC interaction**
  + Short coming of key establishment-symmetric key based
    - No Perfect Forward Secrecy: If the KEK s are compromised, an attacker can decrypt past messages if he stored the corresponding ciphertext
    - Single point of failure: The KDC stores all KEK s. If an attacker gets access to this database, all past traffic can be decrypted.
    - Communication bottleneck: The KDC is involved in every communication in the entire network (can be countered by giving the session keys a long lifetime)
    - Secure channel during initialization: when a new user joins - public key cipher for new key transport
    - A cryptographic protocol has perfect forward secrecy (PFS) if the compromise of long-term keys does not allow an attacker to obtain past session keys. The main mechanism to assure PFS is to employ public-key techniques.
  + key establishment-Asymmetric key based (e.g., DHKE)
    - alpha primitive root
  + Man-in-the-Middle Attack
  + 
  + Certificates and DHKE with Certificates
    - 

 Authentication

* + User Authentication Factors
    - Knowledge-based
    - Possession-based
    - Static biometrics
    - Dynamic biometrics
  + Password Vulnerabilities
    - 
    - User mistakes/multiple password use/electronic monitoring or sniffing
  + Storing passwords
    - Hashing passwords (salting breaks brute force and forward search (RAINBOW ATTACK precomputed hash value table)
  + Password Selection Strategies
    - 
  + Remote user authentication protocols
    - Static: user authenticates herself to the token and then the token authenticates the user to the computer
    - Dynamic password generator: the token generates a unique password periodically (e.g., every minute).
    - Challenge response: Computer system generates a challenge, such as a random string of numbers. The smart token generates a response based on the challenge.
* **Man-in-the-middle** is an *active attack to a cryptographic protocol*, where the attacker is, effectively, in between the communications of two users, and is capable of intercepting, relying, and (possibly) altering messages. In this case, the meaning of "in the middle" is direct: the attacker is in the middle of two communicating users.
* **Meet-in-the-middle** is a type of *cryptanalytic attack* that uses some sort of time-space trade-off to drastically reduce the effort to perform a brute-force attack (e.g., transforming an attack that requires 21282128 time into one that takes 264264 time and 264264 space). In this case, the name of the attack comes from the expression "let's meet in the middle", which means "to make a compromise". It may also refer to a type of attack over certain block ciphers, where the attacker decompose the problem in two halves and proceeds on each part separately.