Crypto Assignment 3

Name: CHE Yulin, Student#: 20292673

Q1 SHA-1 Restriction

- **Answer**: it is enough for hashing any message in the near feature and to attack the SHA-1 with brute force approach is impossible with current computing power. The specific analysis is as follows.
- **Enough Length**: It will take a long time until we reach messages longer than 2^{64} bits, which we want to hash sequentially. To put things in perspective, SHA-1's performance on modern CPUs is about 0.7 cycles/bit. Assuming a 5 GHz clock, it would take 80 years to hash 2^{64} bits. More CPUs do not help.
- **Security Level**: SHA-1 has a security level of 2^{80} bits for brute-force collision attacks. Attackers reach that level of power long before single files reach 2^{64} bits. And the best attacks made by Xiaoyun Wang, Andrew Yao and Frances Yao can find collisions in the full version of SHA-1, requiring 2^{63} operations.

Q2 DSS k-value Leak

- **Answer**: a user's private key is compromised if k-value is compromised.
- **Formula**: two formulas are adopted to generate (r, s) as the signature part in DSS. If a person gets k from the sender, with (r, s) and global public key (p, q, g), x could be deduced.
  $r = (g^k \mod p) \mod q$
  $s = [k^{-1}(h(m) + xr)] \mod q$
- **Deduction**: the deduction procedure is as follows.
  $s = [k^{-1}(h(m) + xr)] \mod q$
  $\rightarrow ks \mod q = (h(m) + xr) \mod q$
  $\rightarrow (ks - h(m)) \mod q = xr \mod q$
  $\rightarrow (ks - h(m)) \cdot r^{-1} \mod q = x \mod q$
- **Conclusion**: thus, we can get $x \mod q$, in formula
  $x \mod q = (ks - h(m)) \cdot r^{-1} \mod q$.
  Here, k, s, m, r, q are all known things.

Q3 Diffe-Hellman Protocol Attack

- **Answer**: the attack procedure is elaborated as follows, namely **session key establishment phase** and **modification or control**, the user Alice is denoted as A, the middle-man is denoted as M, the user Bob is denoted as B.
- **session key establishment phase**:

  1. M generate two random private keys namely X_{M1}, X_{M2}, and then he computes the corresponding public keys namely Y_{M1}, Y_{M2}
  2. Alice generate private key X_A, and computes the corresponding public key Y_A, and then pass Y_A to Bob
  3. M intercepts Y_A, and pass Y_{M1} to Bob. M computes the session key between himself and Alice, in formula $K_2 = (Y_A)^{X_{M2}} \mod q$.
  4. Bob gets Y_{M1} and computes session key in formula $K_1 = (Y_{M1}^{X_B}) \mod q$
  5. Bob pass Y_B to Alice
  6. M intercepts Y_B, and pass Y_{M2} to Alice. M computes session key between himself and Bob, in formula $K_1 = (Y_B)^{X_{M_1}} \mod q$
  7. Alice gets Y_{M2}, and computes session key in formula $K2 = (Y_{M2})^{X_A} \mod q$

After session key establishment phase, Bob and Alice assumes that they share the session key with each other. However, in fact, Bob and M shares K_1 while Alice and M sahres K_2.

- **message modification or control phase**
  1. Alice sends a message to Bob, $E(K_2, M)$
  2. M intercepts the message, decrypts the message with session key K_2 and gets the message M
  3. M sends $E(K_1, M)$ without modification or sends $E(K_1, M^{changed})$ with modification to Bob

In message modification or control phase, M is able to either modify the message or simply intercept the message.

- **weakness**: this lies in the lack of authentification of communicators, which could be fixed with the introduction of digital signature and public key certificate.

Q4 Kerberos Key Classificatin

- **Answer**: there are three types of secret keys, corresponding to three phases in Kerberos authentification dialogue, namely authentification service exchange, ticket-granting service exchange and client/server authentification exchange.
- **Classification on Generated Time**
  - **keys established before**: K_c denotes client's token shared between the client and AS, K_{tgs} denotes the secret key shared between AS and TGS, K_v denotes the secret key shared between TGS and V

- **keys established In dialogue**: $K_{c,tgs}$ denotes the secret key shared between the client and TGS, $K_{c,v}$ denotes the secret key shared between the client and V

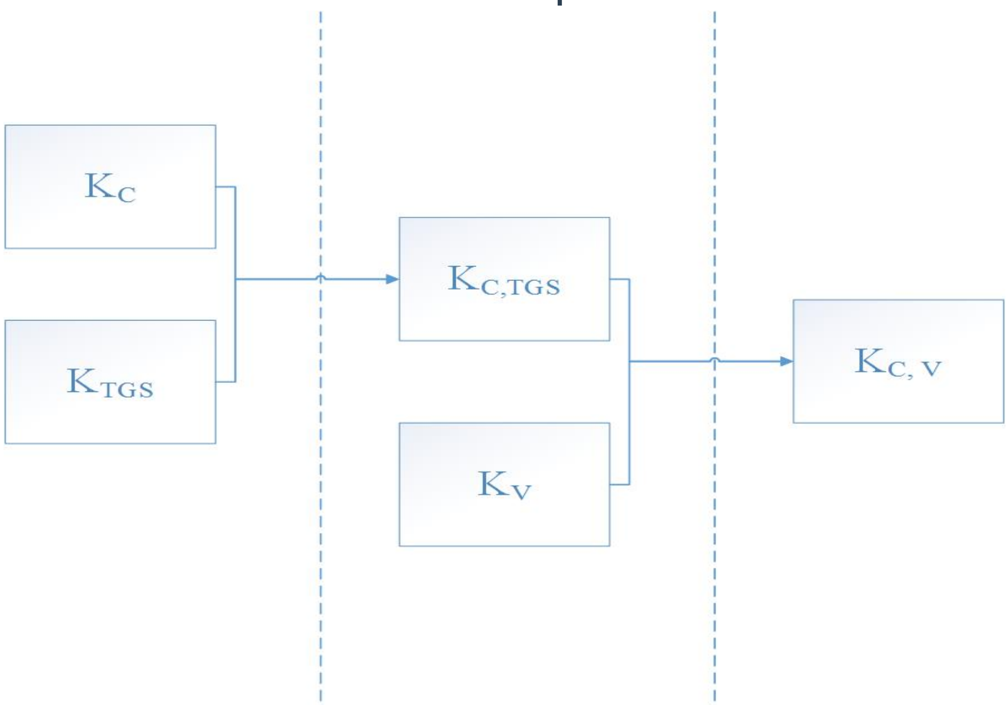
- **Classification on Three Phases for Authentification**
  In each phase, the keys used to keep the authentification of entities of communications are pointed out.

  - **obtaining ticket-grant ticket phase** K_c
  - **obtaining service-grant ticket phase**: $K_{c,tgs}$
  - **obtaining service phase**: $K_{c,v}$

- **Classification on Two Phases for Confidentiality for Newly Generated Keys**

  - **establishing** $K_{c,tgs}$ **phase**: K_c, K_{tgs}
  - **establishing** $K_{c,v}$ **phase**: $K_{c,tgs}$, K_v

- **Hierarchical Protection Pictorial Description**



Q5 Kerberos Authentification

- **Answer**: This is to prevent the use of the tickets from workstation other than the one that initially requested them.
- **Elaboration**: if there is no network address of C, attackers are able to intercept the ticket and used identity ID_C to send messages to server V, from other workstations which are different from C.
- **Attack Example**: if there is no network address of C, in client-server authentification phase, the attacker is able to intercept C's message $Ticket_v || Auth_c$, and sends it to sever to obtain the service from V.

Q6 SSL vs IPsec

- **Answer**: The reason mainly lies in that SSL is built upon TCP transport protocol with states and connections which requires synchronizations, while IPsec is much deeper without guaranteeing connections.
- **SSL**: The SSL session is bidirectional and thus stateful. Hence synchronization is necessary. The change cipher spec protocol is just for this purpose.
- **IPSec**: On thecontrary, the IPSec association is for one direction only and is not stateful, and there is thus no need to do the synchronization. Hence IPSec does not need such a protocol.