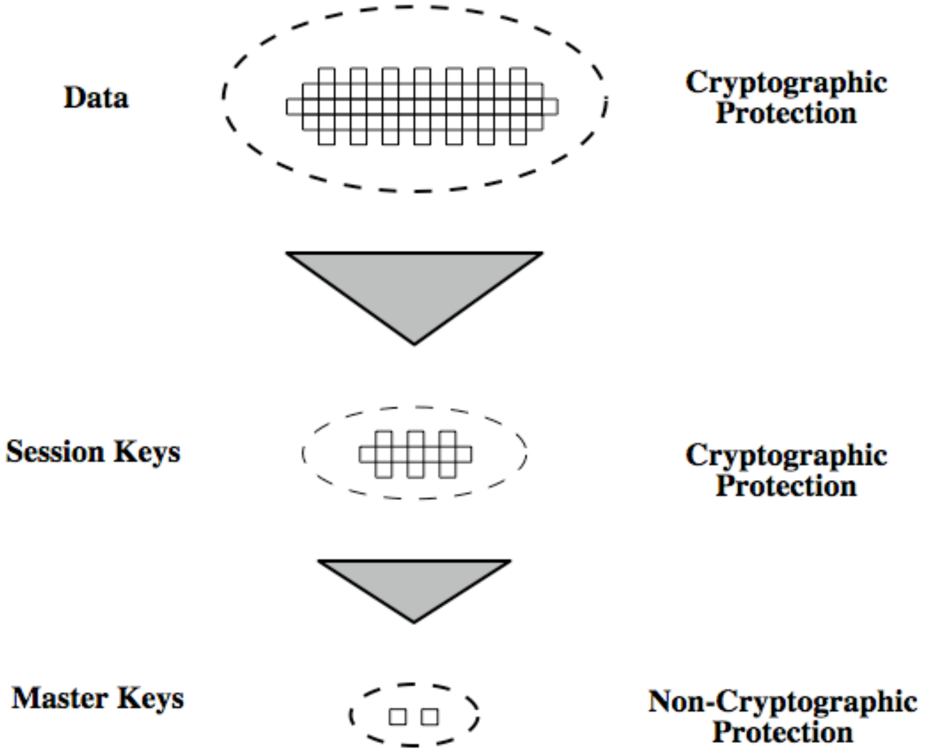
Key Hierarchy

- ❑ Usually there is a key hierarchy
  - ➢ Master key/secret (key encryption key)
    - o used to establish/distribute session keys
  - ➢ Session key (data encryption key)
    - o used to encrypt data/message
    - o for one logical session only



Session Keys

- ❑ More often a symmetric key is used, more likely it may be compromised.

- ❑ Generate and use a symmetric (secret) key for one session only → session key.

- ❑ Using different session keys in different sessions can
  - ➢ limit available ciphertexts for cryptanalysis.
  - ➢ limit exposure (both in time period and amount of data) in an event of key compromise.

- ❑ To avoid long-term storage of a large number of secret keys, we only generate them when they are needed.

Session Key Establishment

- ❑ Session key establishment solutions
  - ➢ Key agreement (exchange) protocols
    - o A shared secret (master or session secret) is derived by the parties as a function of information contributed by each, such that no party can predetermine the resulting value - Diffie-Hellman (DH) protocol.
  - ➢ Key transportation/distribution protocols
    - o Without any use of a public-key cipher (PKC)
      - • Session keys are generated and distributed using symmetric-key cipher and with the help of a third party - the Needham-Schroeder protocol.
    - o With the use of a public-key cipher
      - • One party creates a secret value (session key), and securely transfers it to the other party using the recipient’s public key.

Session Key Establishment

- ❑ There are other issues that should be considered
  - ➢ Key secrecy and entity/key authentication
    - o Assurance: no other party (outsiders - apart from the entities involved) could gain access to the established session key.
    - o The session key is established with the intended entities.
    - o Key confirmation: asking the other entity (possibly unidentified) to demonstrate that he has the knowledge of the key by
      - • producing a one-way hash value of the key; or
      - • encrypting some known data (e.g. nonce) with the key.
  - ➢ Key freshness
    - o Assurance: the key is fresh, i.e. not used before.