Computer and Network Security: Authentication Protocols: Cryptographic Authentication

Kameswari Chebrolu

Outline

- ~~Human Authentication~~
  - ~~Focus: Password based systems~~
- **Cryptographic Authentication (Human as well as computer)**
  - **One way authentication (shared and public key)**
  - Mutual authentication (shared and public key)
  - Mediated authentication (shared key)
  - How to incorporate session key exchange?
  - How to follow it up to provide privacy and Integrity?

Players

- Lovers: Alice (A) and Bob (B)
  - In computer world: web browser/server; bank client/server; routers etc
- Enemy: Mallory (M, malicious)
- Supporting cast: KDC, CA
- Assumption: Long-term keys in place

Role of Passwords

- Crypto → keys → large random bit string
  - Computer-computer interaction ok
  - Human-computer? Humans not good at remembering bits
- Derive keys from passwords
  - Hash of password
    - Works for secret key, not straightforward for public key
  - Use password to decrypt an encrypted key
  - Note: poor passwords can be cracked offline

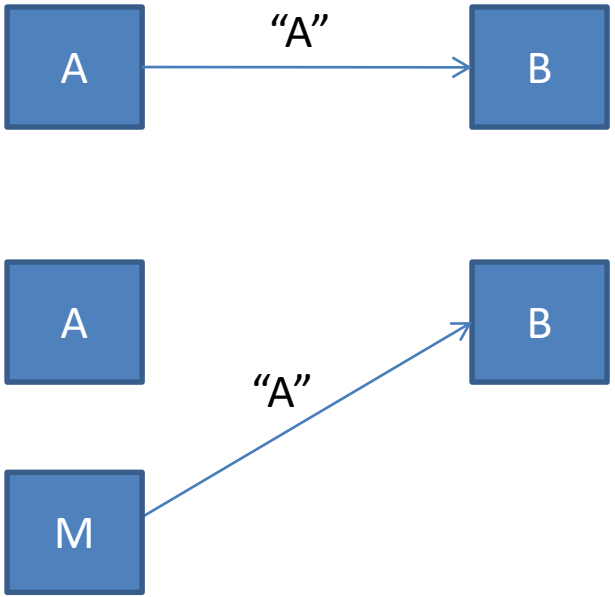
Storage of Keys

- Store passwords on machine encrypted with strong human passwords
- Removable media
- Tamper resistant devices (accessible only via embedded software)

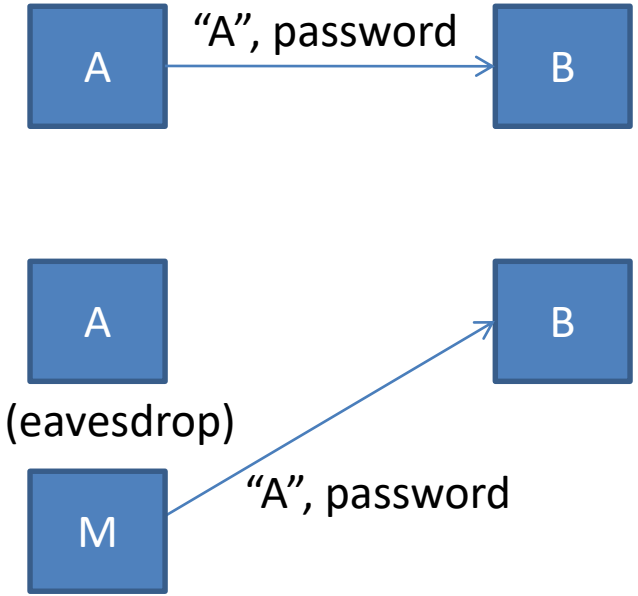
One-way Authentication

A is proving to B that it is indeed A

- APv1

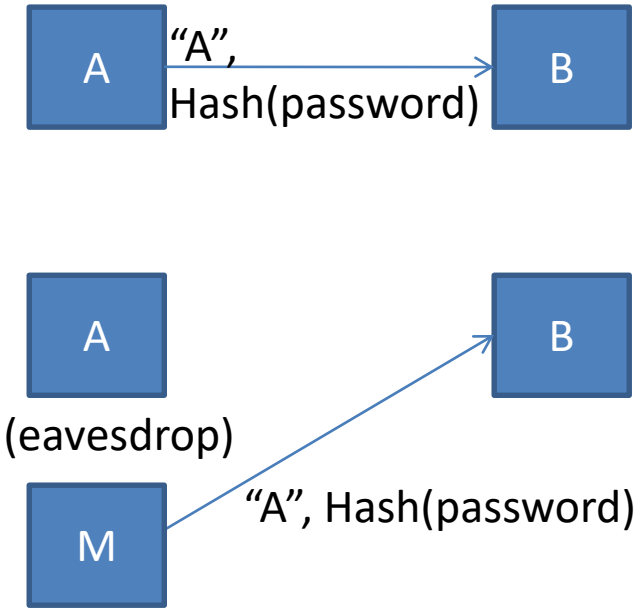


- APv2



APv3: Crypto

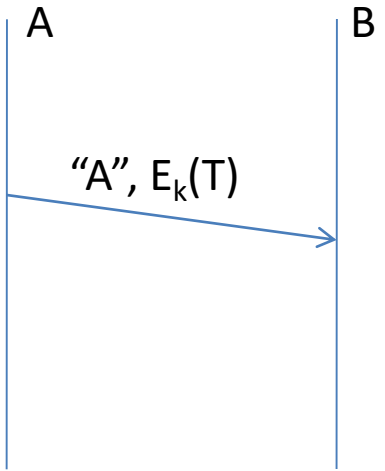
- M does not know the password but can replay message
  - Same as APv2
- Timeliness/Originality concern



Possible Solutions

- Message is indeed from A (1) and further not a replay (2)
- (1) can be ensured via an authenticator (MAC/signature)
- For (2): How about (tamperproof) timestamps?

- Value of the timestamp must be within an acceptable range of the current time
- Replay?
  - Timestamp is out of range
- Replay possible within time window
- Requires clock synchronization; tough in practice

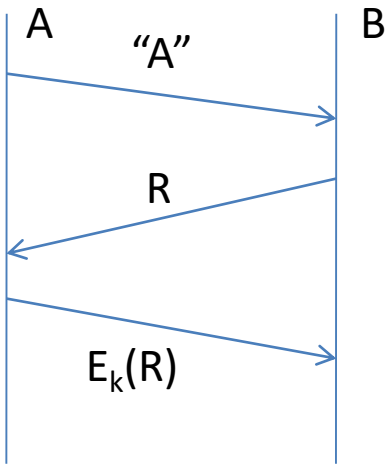


Possible Solutions

- Message is indeed from A (1) and further not a replay (2)
- (1) can be ensured via an authenticator (MAC/signature)
- For (2): How about use of (tamperproof) nonce (random number used only once)
    - Requires keeping track of all past nonces

Challenge-response

- Use challenge response in combination with nonce
  - Keep track of only nonces whose response outstanding
  - Nonce is 256 bits
    - Prob of choosing same nonce twice is infinitesimally small

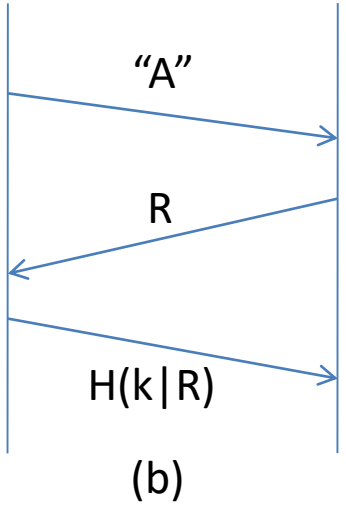


(a)

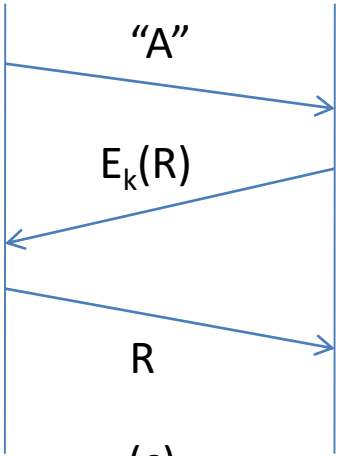
Using secret/shared key

Other Variants

- (b) may be faster (hash is simpler than AES) than (a)
- (c) is similar to (a)
- In (c) can Encryption be replaced by hash function?
  - No. Function needs to be reversable



(b)

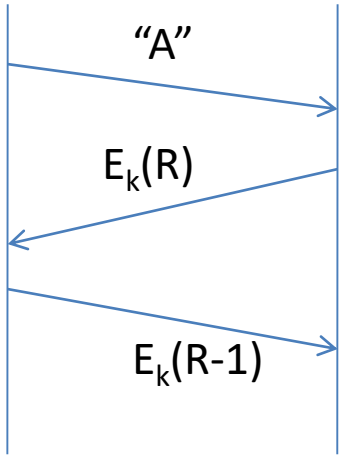


(c)

Using secret/shared key

Yet Another Variant

- Does not reveal message-ciphertext pairs for crypto-analysis



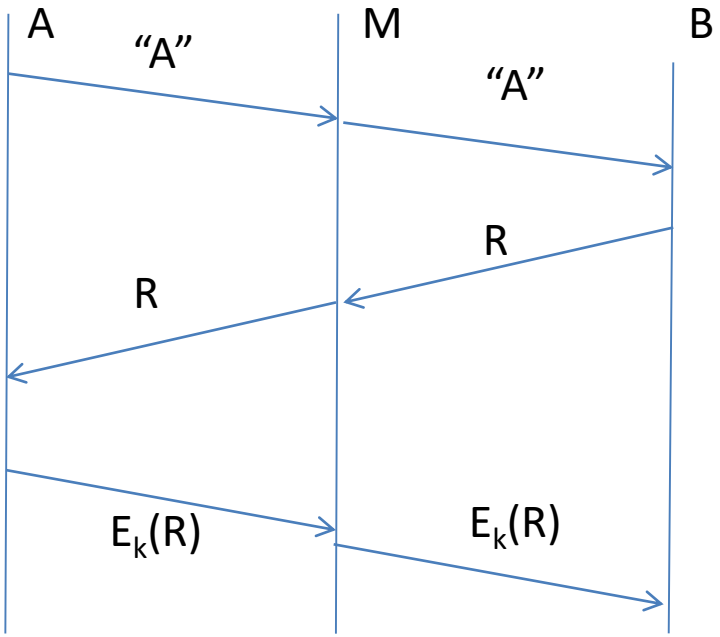
(d)

Points to Note

- It is one way authentication: B is authenticating A
  - A is not authenticating B → Attacker(M) can trick A into believing it is B; M can send R and ignore A's response
- After authentication, can remaining message exchange be in open?
  - Attacker can take over the conversation

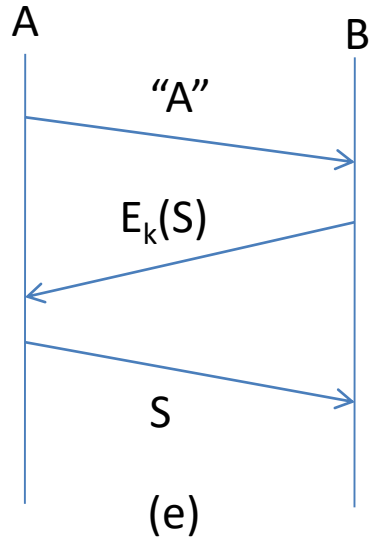
Points to Note

- Man in the middle attack?
  - Possible in most authentication protocols
  - E.g. a rouge router
  - Difficult to achieve in practice
  - Important to cryptographically protect later messages

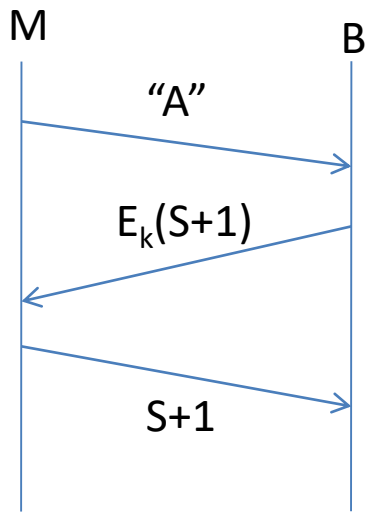


Can R be predictable?

- Use sequence numbers (S)?
  - Require non-volatile state to handle crashes
- (e) is not secure



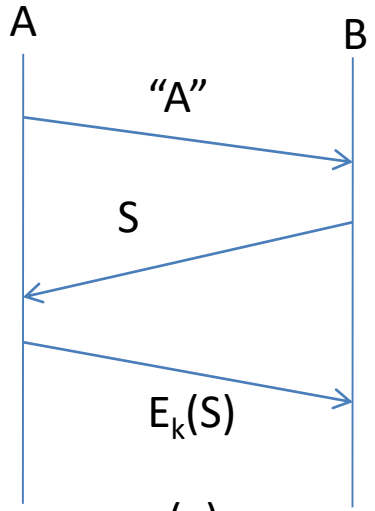
(e)



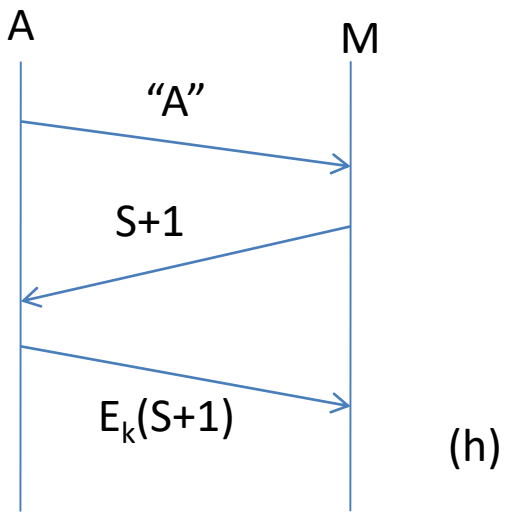
(f)

Can R be predictable?

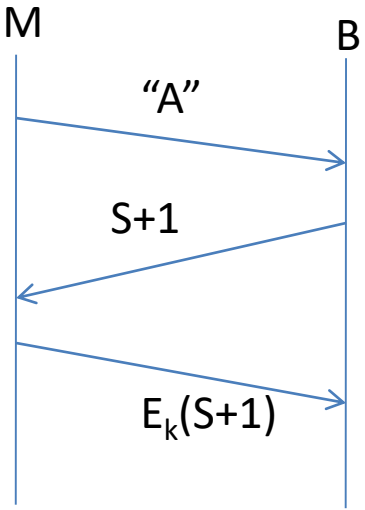
- (g) is also not secure
- Large random number best as nonce



(g)



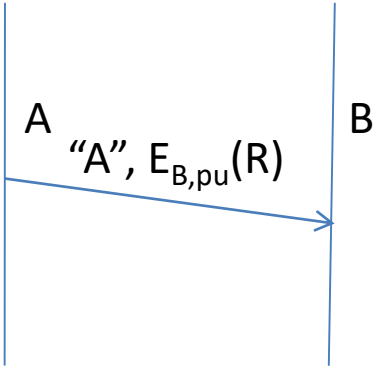
A is not authenticating M



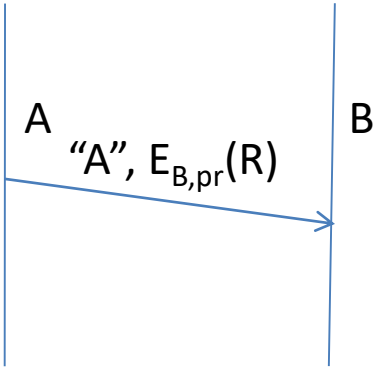
(h)

Asymmetric Key based Solutions

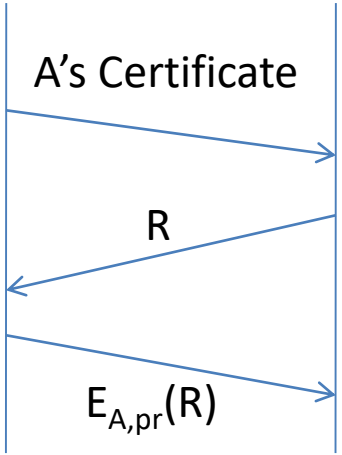
- (a) is not secure
  - M can easily construct the message too
- (b) ok but requires keeping track of all nonces
- Similar issues as symmetric key approaches



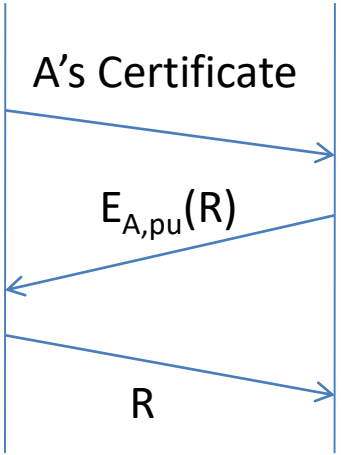
(a)



(b)



(i)



(j)

Using asymmetric/public key

Note: You can use nonces to trick some one to sign or decrypt messages

Solution: Use different keys for different purposes (or)

R should have structure: type field concatenated with data (PKCS standard)

Summary

- One-way authentication: Message is indeed from A and further not a replay
  - MITM difficult to prevent in most authentication protocols
- Challenge-Response with “random” nonce best
  - Handles authentication and replay
  - Can be implemented using shared (see a,b,c,d) and public key (see i,j)