Computer and Network Security: Mutual Authentication

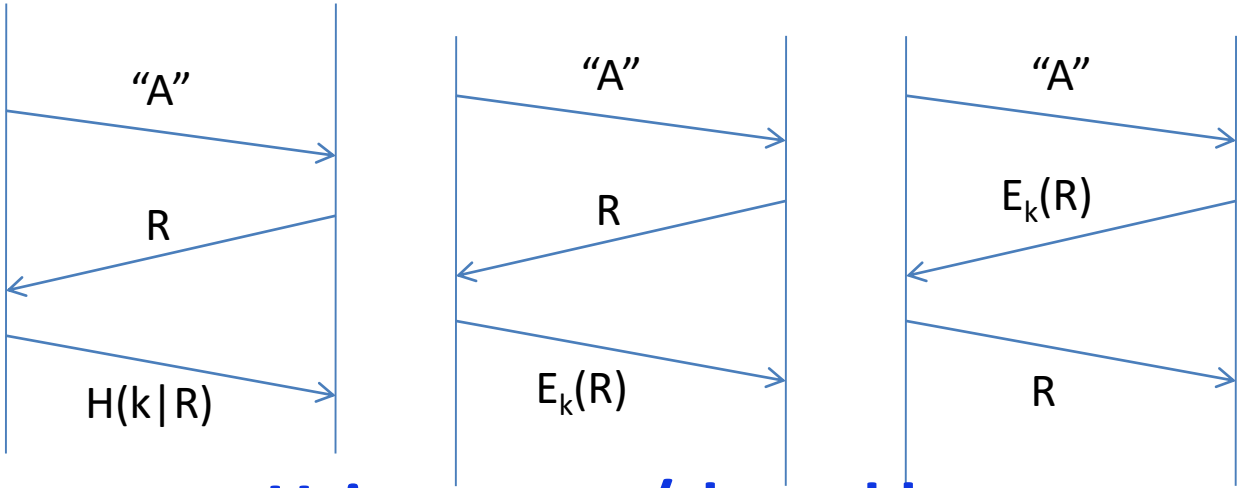
Kameswari Chebrolu

Outline

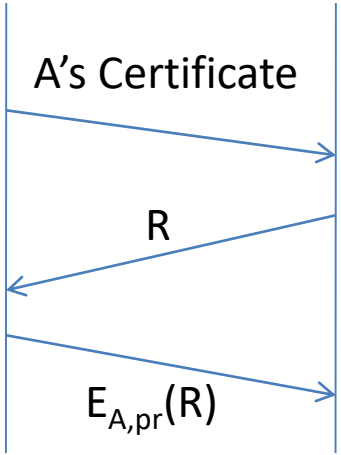
- Human Authentication
  - Focus: Password based systems
- **Cryptographic Authentication** (Human as well as computer): Prove identity by performing a cryptographic operation (hash, encryption etc)
  - ~~One way authentication (shared and public key)~~
  - **Mutual authentication (shared and public key)**
  - **How to incorporate session key exchange?**
  - **How to follow it up to provide privacy and Integrity?**
  - Mediated authentication (shared key)

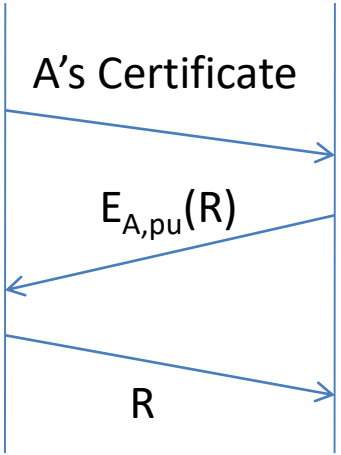
Recap: Challenge-response

- 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



Using secret/shared key



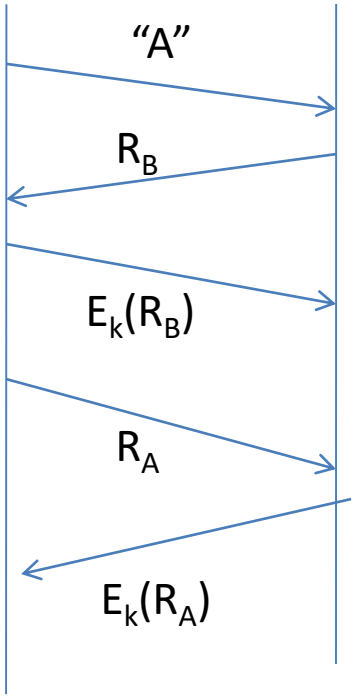


Using asymmetric/public key

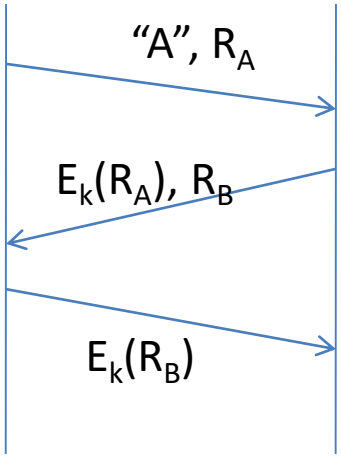
Note: You can use nonces to trick some one to sign or decrypt messages
Solution: R should have some structure for different uses

Mutual Authentication: Shared key

- (A) is secure but not (B)
- Why?

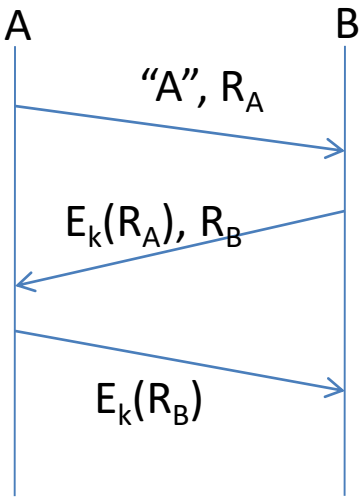


(A)

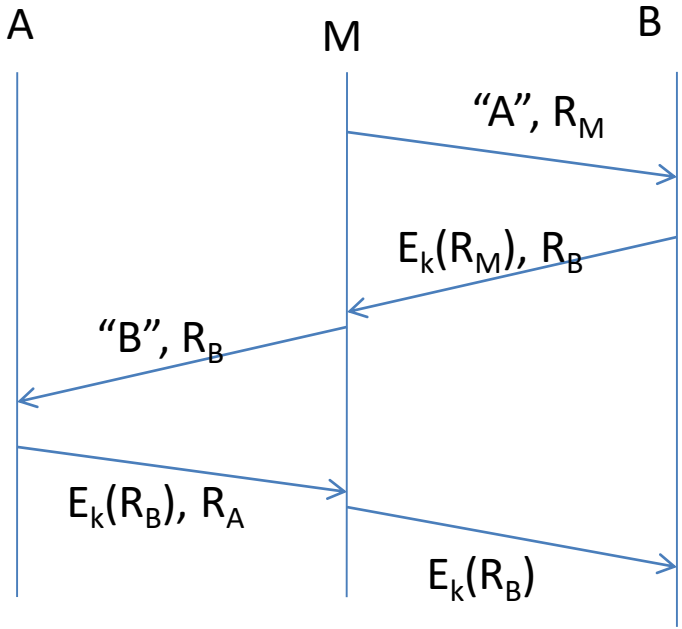


(B)

Reflection Attack

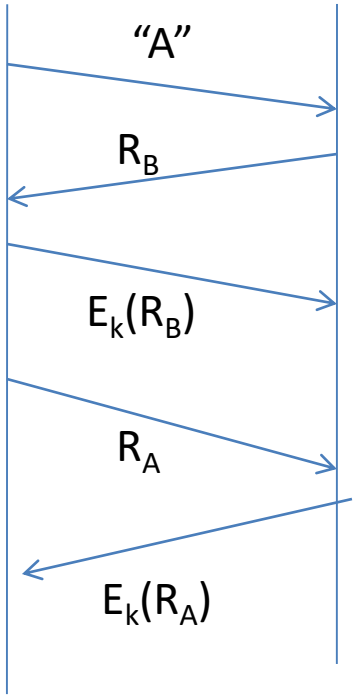


Flawed Protocol



Attack

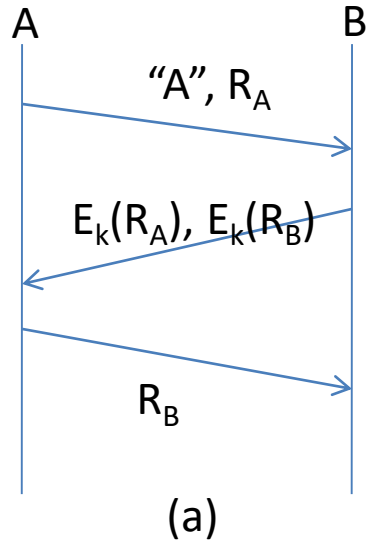
- Why does this not occur in (A)?
  - Good practice to let initiator prove identity first before responding



(A)

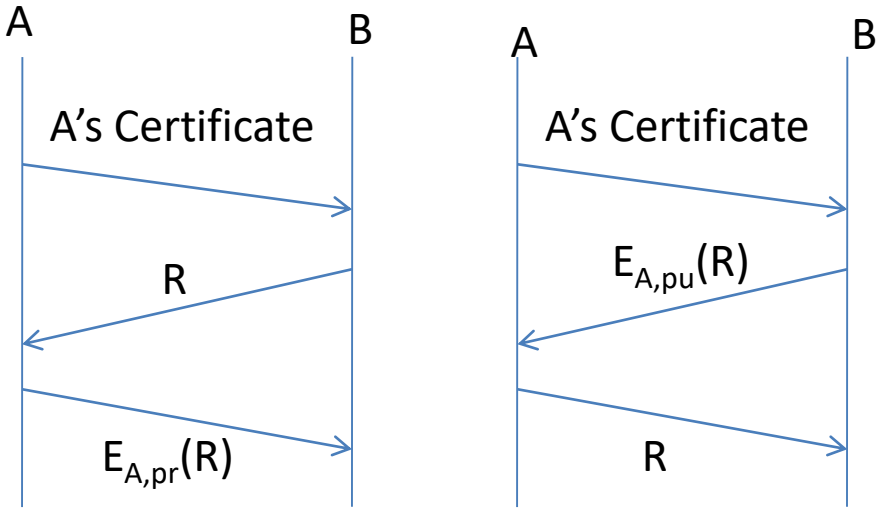
Other Solutions

- Initiator/responder draw challenges from different sets
  - Initiator challenge odd number; respond challenge even number
- Responder to encrypt; initiator to decrypt (see (a))

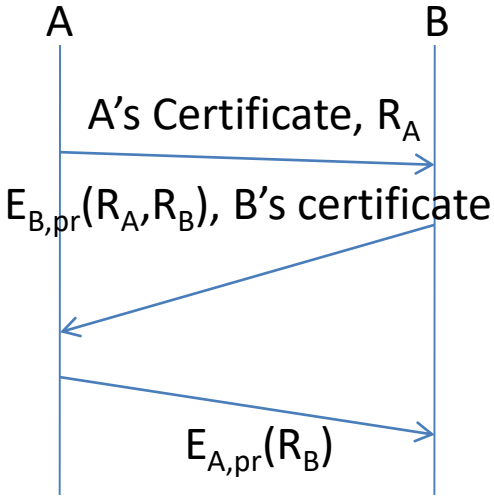


(a)

Public Key Authentication



(a) One-way



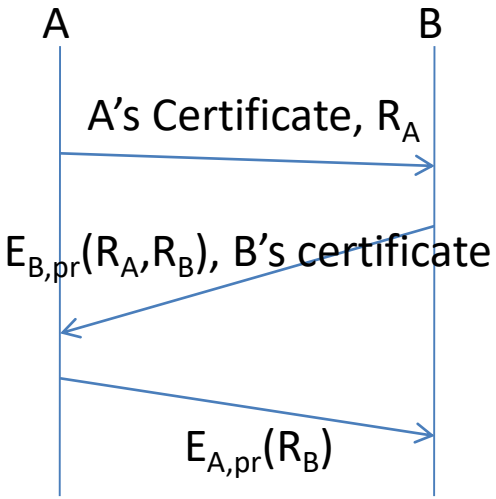
(b) Two-way ?

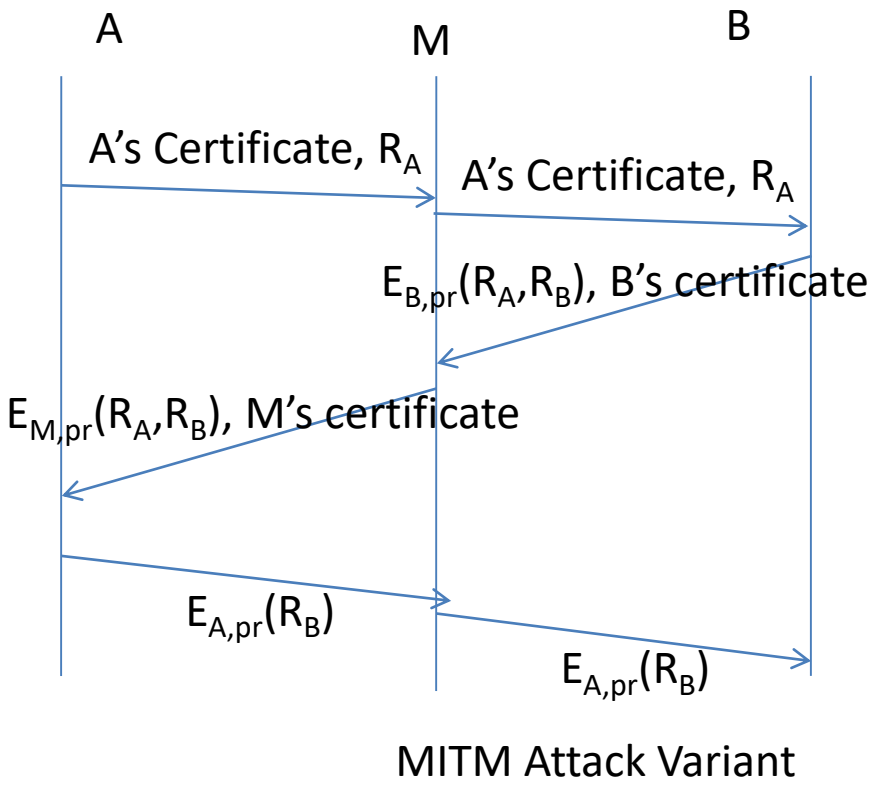
Reflection Attack and Solution

- Does it apply here?
- If so, what solution?

(Homework)

Mutual Authentication: Public key

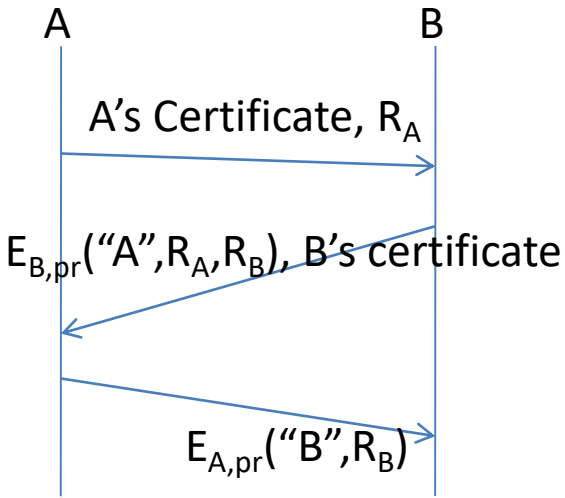




MITM Attack Variant

Solution

- In signed messages, include identity of recipient



Correct Protocol

Summary

- Mutual authentication: Both parties check if message is indeed from the other party and further not a replay
  - Challenge-response based
  - Need attention to detail
- Reflection/MITM attacks possible
- Best practices:
  - Draw challenges from different sets
  - Include identity of recipients in the proof

Outline

- Human Authentication
  - Focus: Password based systems
- **Cryptographic Authentication** (Human as well as computer): Prove identity by performing a cryptographic operation (hash, encryption etc)
  - ~~One way authentication (shared and public key)~~
  - ~~**Mutual authentication (shared and public key)**~~
  - **How to incorporate session key exchange?**
  - **How to follow it up to provide privacy and Integrity?**
  - Mediated authentication (shared key)

How to incorporate session key?

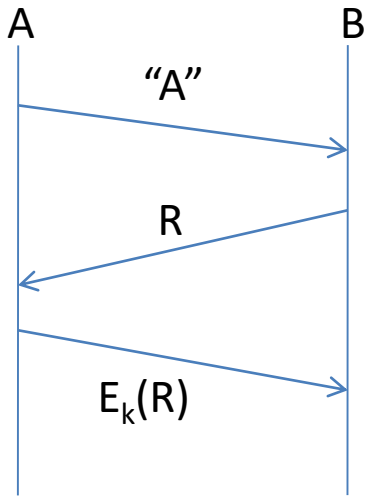
- Post authentication, desirable to have short-term session key for confidentiality/integrity
  - This key valid for this session alone

Secret Key Crypto

- End of authentication
  - long-term key k;
  - nonce R (one-way) or R_A/R_B (mutual)

Focus: One-way

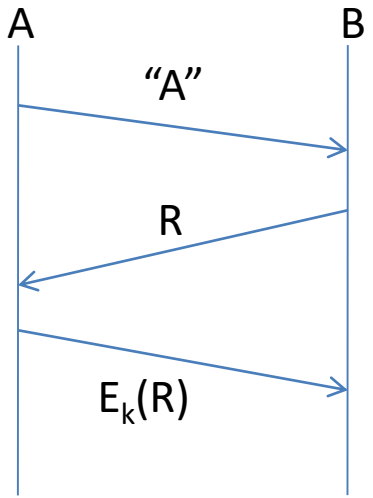
- Can the session key be $E_k(R)$?
  - It goes in the open



One way secret key based authentication

Secret Key Crypto

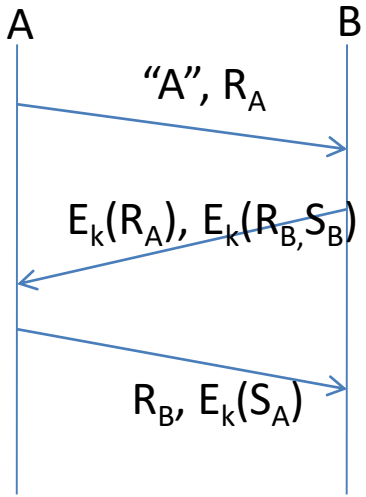
- Can the session key be $E_k(R+1)$?
  - M can act as B and throw (R+1) as challenge to A
- Some combination of k,R can be used as session key but not all acceptable
  - $E_{k+1}(R)$ for one-way or $E_{k+1}(R_A)$ for two-way
  - or $E_{K+1}(R_A) \oplus E_{k+1}(R_B)$ for two-way



One way secret key based authentication

- Another Solution
- What makes a good session key?
  - Different for each session
  - Un-guessable
  - Not made up of X, where X is predictable and encrypted or signed by long-term key
  - In case of mutual authentication, preferable if both contribute to it

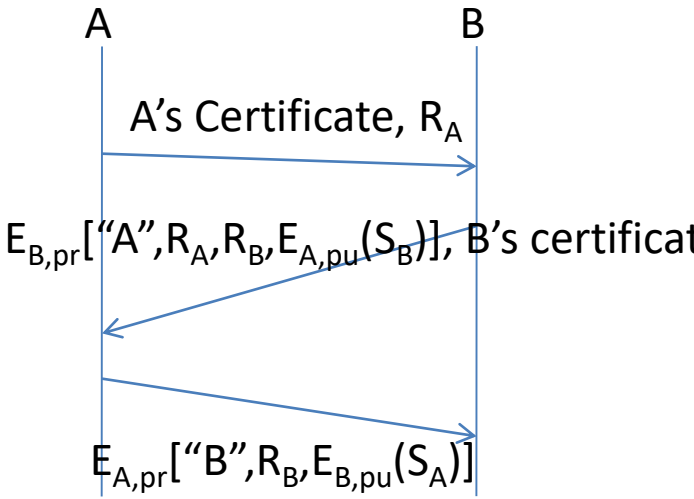
$$\text{Session key} = S_A \oplus S_B$$



Session key for mutual auth based on secret key crypto

Public Key Crypto

- Similar in spirit to secret-key crypto (solution-2)
- Session keys S_B and S_A need not be signed
  - No loss of security as such



$$\text{Session key} = S_A \oplus S_B$$

Use of short-term session key

- Provides confidentiality and integrity
- No standard algorithm to do both in one single cryptographic pass (refer to MACs)
- Possibilities:
  - Use two keys for two operations
    - During authentication phase, exchange two session keys
    - Or derive the other key from the exchanged key
  - Use weaker checksum for integrity inside stronger confidentiality

- Replay attacks/MITM can still disrupt sessions
- Best Practices
  - Use of sequence numbers to order messages
  - Integrity check: function of all previous messages
  - For bi-directional communication (to avoid reflection attacks)
    - Use sequence numbers in different ranges
    - Use different integrity algorithms in each direction
  - Change session keys periodically during conversation (key rollover)
- Will explore more of these practices in SSL/TLS

Summary

- Mutual authentication: Both parties check if message is indeed from the other party and further not a replay
  - Challenge-response based
  - Need attention to detail
- Reflection/MITM attacks possible
- Certain best practices mitigate some of these attacks
- Post authentication
  - Derive session key as a function of long-term key and nonces
  - (or) derive final session key from encrypted session keys sent during authentication