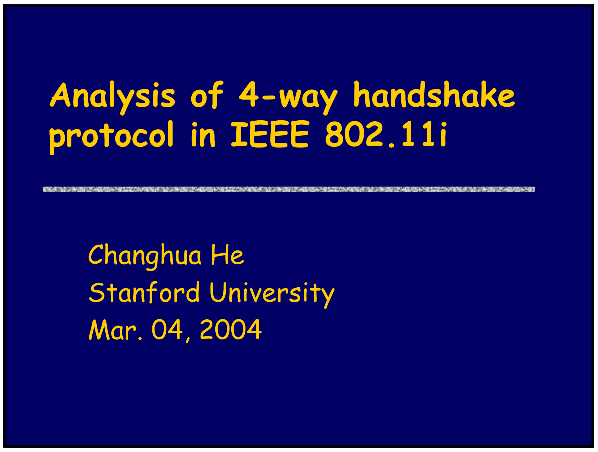
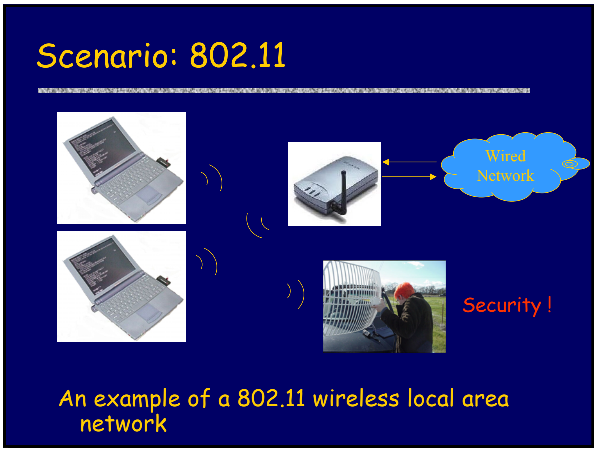


Analysis of 4-way handshake protocol in IEEE 802.11i

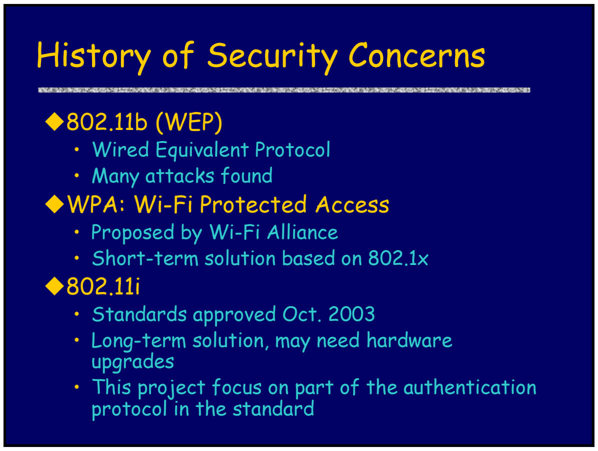
Changhua He
Stanford University
Mar. 04, 2004



Scenario: 802.11

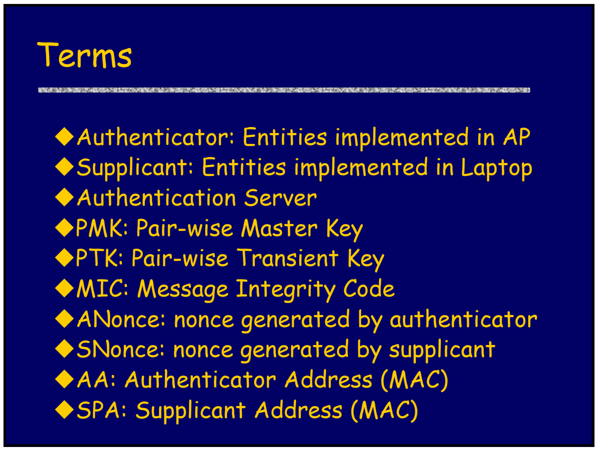
Security !

An example of a 802.11 wireless local area network



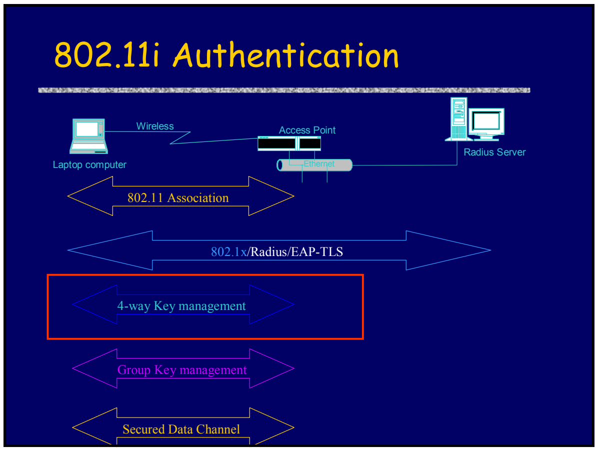
History of Security Concerns

- 802.11b (WEP)
  - Wired Equivalent Protocol
  - Many attacks found
- WPA: Wi-Fi Protected Access
  - Proposed by Wi-Fi Alliance
  - Short-term solution based on 802.1x
- 802.11i
  - Standards approved Oct. 2003
  - Long-term solution, may need hardware upgrades
  - This project focus on part of the authentication protocol in the standard



Terms

- Authenticator: Entities implemented in AP
- Supplicant: Entities implemented in Laptop
- Authentication Server
- PMK: Pair-wise Master Key
- PTK: Pair-wise Transient Key
- MIC: Message Integrity Code
- ANonce: nonce generated by authenticator
- SNonce: nonce generated by supplicant
- AA: Authenticator Address (MAC)
- SPA: Supplicant Address (MAC)



802.11i Authentication

Laptop computer

Wireless

Access Point

Ethernet

Radius Server

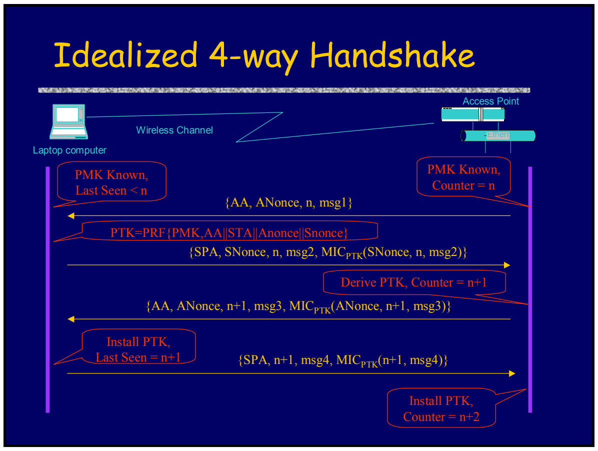
802.11 Association

802.1x/Radius/EAP-TLS

4-way Key management

Group Key management

Secured Data Channel



Idealized 4-way Handshake

Laptop computer

Wireless Channel

Access Point

Ethern

PMK Known,
Last Seen < n

PMK Known,
Counter = n

{AA, ANonce, n, msg1}

PTK=PRF{PMK,AA||STA||Anonce||Snonce}

{SPA, SNonce, n, msg2, MIC_{PTK}(SNonce, n, msg2)}

Derive PTK, Counter = n+1

{AA, ANonce, n+1, msg3, MIC_{PTK}(ANonce, n+1, msg3)}

Install PTK,
Last Seen = n+1

{SPA, n+1, msg4, MIC_{PTK}(n+1, msg4)}

Install PTK,
Counter = n+2

Description

- Prior to 4-way handshake, we assume:
  - PMK only known to Supplicant and Authenticator, never transmitted over network
- Objectives:
  - Generate PTK and confirm the procession and freshness of PTK
- Methodology:
  - Use Murφ to model the protocol from simplest version, find out attacks, add fields step by step to defense the attacks, get complete one.
  - Can make clear the function of each fields, and find out attacks for the complete protocol.

Murφ Modeling

- Authenticators/Supplicants:
  - Each authenticator maintain associations with each supplicant, and vice versa
  - Each association has a unique PMK
  - Several sessions can happen in one association sequentially
- In each run:
  - Turn on/off fields: nonce, sequence, mtype, address

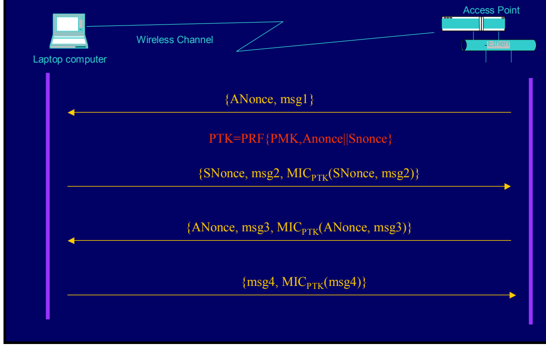
Intruder

- Impersonate both supplicant and authenticator
  - Forge MAC address in each message
  - Can not get PMK for associations
- Intercepts all messages
- Replay all messages
- Forge messages with known nonce and MIC
- Compose message 1 with known nonces
- Actively predict nonces and ask the supplicant to pre-compute MIC
  - Model attacks when nonces are predictable or not globally unique

Invariant

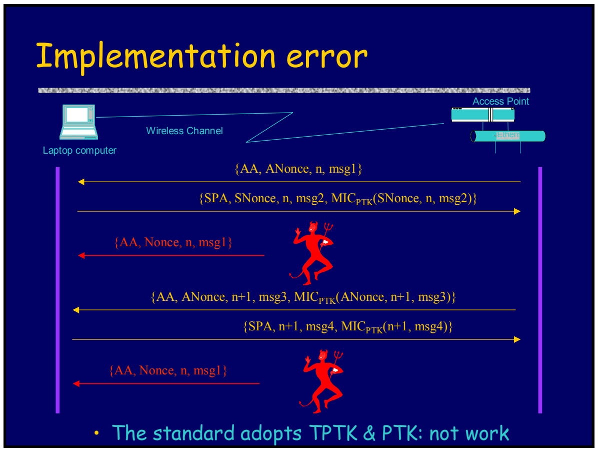
```
invariant "PTKs are consistent and fresh"
forall i: AuthenticatorId do
  forall j: SupplicantId do
     aut[i].associations[j].session.state = A_DONE

     ->

     (sup[j].associations[i].session.state = S_DONE    &
       ptkEqual(aut[i].associations[j].session.ptk,
                   sup[j].associations[i].session.ptk)       &
       aut[i].associations[j].sid = sup[j].associations[i].sid)  |
     (sup[j].associations[i].session.state = S_PTKSA   &
       aut[i].associations[j].sid <= sup[j].associations[i].sid)
  end
end;
```

Achieved protocol


Access Point
Laptop computer
Wireless Channel
{ANonce, msg1}
PTK=PRF{PMK,Anonce||Snonce}
{SNonce, msg2, MIC_{PTK}(SNonce, msg2)}
{ANonce, msg3, MIC_{PTK}(ANonce, msg3)}
{msg4, MIC_{PTK}(msg4)}

Summary of fields

- Nonces is necessary for fresh PTK
- Mtype
  - Necessary, otherwise can fool supplicant to calculate msg 3, or vice versa
- Sequence
  - Not necessary here
  - Defense msg 3 replay, but it is harmless
- AA, SPA
  - Bind PTK to the physical device, not necessary here, but need to be considered with PMK



Implementation error

{AA, ANonce, n, msg1}

{SPA, SNonce, n, msg2, MIC_{PTK}(SNonce, n, msg2)}

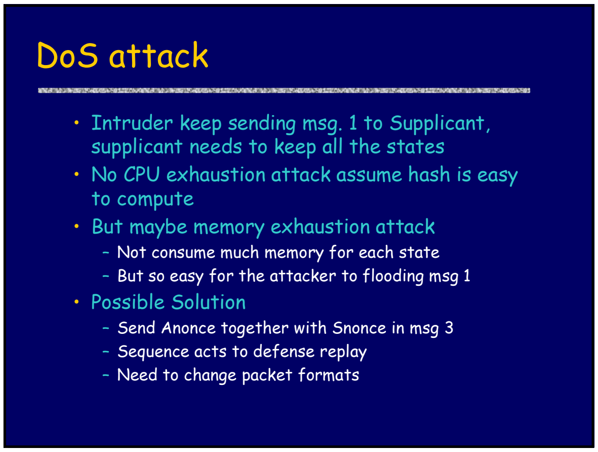
{AA, Nonce, n, msg1}

{AA, ANonce, n+1, msg3, MIC_{PTK}(ANonce, n+1, msg3)}

{SPA, n+1, msg4, MIC_{PTK}(n+1, msg4)}

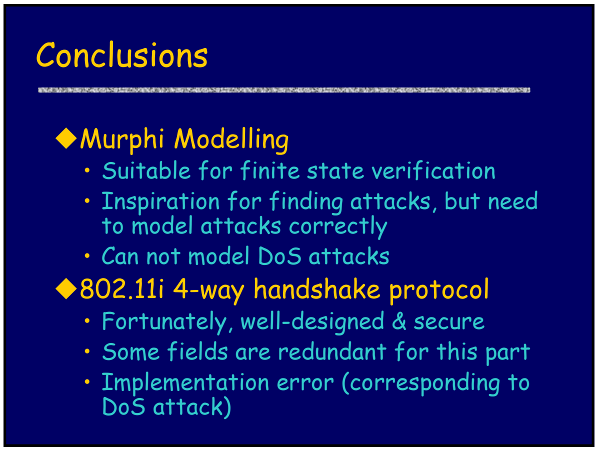
{AA, Nonce, n, msg1}

- The standard adopts TPTK & PTK: not work



DoS attack

- Intruder keep sending msg. 1 to Supplicant, supplicant needs to keep all the states
- No CPU exhaustion attack assume hash is easy to compute
- But maybe memory exhaustion attack
  - Not consume much memory for each state
  - But so easy for the attacker to flooding msg 1
- Possible Solution
  - Send Anonce together with Snonce in msg 3
  - Sequence acts to defense replay
  - Need to change packet formats



Conclusions

- ◆Murphi Modelling
  - Suitable for finite state verification
  - Inspiration for finding attacks, but need to model attacks correctly
  - Can not model DoS attacks
- ◆802.11i 4-way handshake protocol
  - Fortunately, well-designed & secure
  - Some fields are redundant for this part
  - Implementation error (corresponding to DoS attack)