### Csci388 Wireless and Mobile Security – Key Hierarchies for WPA and RSN

**Xiuzhen Cheng**  
*cheng@gwu.edu*

#### TLS Basics – Revisited

Choose secret $S$

- Compute $K = f(S, R_{Alice}, R_{Bob})$

A key

Choose secret $S$

- Compute $K' = f(S, R_{Alice}, R_{Bob})$

I want to talk, Ciphers I support, $R_{Alice}$

Certificate, cipher I choose, $R_{Bob}$

$\{Certificate\}$

$\{Key\}$

Finish!

Data protected with keys derived from $K$

#### WPA and RSN Key Hierarchy

- **Pairwise Key Hierarchy**
  - **Pairwise Master Key (PMK):** Either preshared or delivered from the upper-layer authentication
  - **PMK is the top of the pairwise key hierarchy**
  - One PMK for each mobile device, shared with the Authentication Server, from which all other pairwise keys are derived
  - **PMK generated at the authentication server**
    - Authentication needs a "supreme secret", which is different than PMK
    - Authentication procedure generates a PMK shared by the server and the supplicant
    - Transferring the PMK from the server to the AP needs protection
      - 802.11i does not specify how RADIUS if the server and the AP do not collocate – specified in WPA; RADIUS has an attribute for this purpose

- **Group Key Hierarchy**

- **Key Derivation**

#### Terminologies

- **Pairwise Key:** protect the communication between an access point and a mobile station
- **Group Key:** shared by a trusted group containing multiple parties
- **Pairwise Key Hierarchy:** all the keys used between a pair of devices (one of which is usually the access point)
- **Group Key Hierarchy:** Various keys shared by all the devices in the group.
- **Preshared keys:** keys installed in the access point and in the mobile device by some method outside WPA/RSN
  - WEP uses preshared keys, possession of the key means authenticity
- **Server-Based keys:** generated by the upper layer authentication protocol such as TLS

#### Review on 802.1X Access Control

- **Association Request**
  - Association Response
  - (RADIUS packet)
  - Request Identity
  - Response Identity
- **Request/Method**
  - **Response/Method**
  - **EAP-Success**
  - **EAPOL-key**
  - **Data**
  - **…**
  - **EAPOL-Logoff**
**Pairwise Master Key**

- **PMK is required to be 256 bits long**
  - Can you memorize the 32 bytes preshared PMK? – use a shorter password, as suggested by the 802.11i
- **PMK is not used directly for any security operations**
  - Temporal keys are generated from PMK
  - Temporal keys are recomputed when a mobile device associates to the access point
  - Two sets of temporal keys: one for EAPOL handshake and one for data
  - All temporal keys must be 128 bits in length
  - All temporal keys form the pairwise transient key (PTK)

**Temporal Keys**

- **Four temporal keys**
  - Data Encryption Key (128 bits)
  - Data Integrity Key (128 bits)
  - EAPOL-Key Encryption Key (128 bits)
  - EAPOL-Key Integrity Key (128 bits)
- **Need liveness to make sure that every recomputation generates a different set of keys**
  - Nonces for liveness
  - MAC addresses for binding the keys with the identity of the devices

**Authenticating the Access Point**

- **Authenticator == Access Point**
- **Supplicant == Mobile device**
- **Mobile devices have to verify the access point**
  - Access point and a mobile device prove to each other that they own the PMK key
  - Through a four-way handshake protocol with the EAPOL-Key message
- **Needs a shared key between the access point and the authentication server**
  - PMK is computed by the server and the supplicant
  - AP receives PMK from a server through a secure channel

**Four-Way Handshake**

- **Authenticator generates ANonce; Supplicant generates SNonce**
- **Four EAPOL-Key messages (unencrypted) are involved**
  - Msg C and D are for synchronization – install keys simultaneously
- **All temporal Keys will be effective after this handshake**

**Group Key Hierarchy**

- **Group key needs rekeying when membership change**
- **Wait until pairwise keys are available then send group keys**
- **At the Access Point:**
  - Create a 256-bit group master key (GMK)
  - Derive the 256-bit group transient key (GTK) from which the group temporal keys are obtained
  - After each pairwise secure connection is established
    - Send GTK to mobile devices through an EAPOL-Key message
    - Check for ACK of the receipt

**How to update group keys without breaking the service? – group key delivery takes time**

- WEP provides the place (identified by the KeyID field) for 4 keys to be stored simultaneously
- Pairwise key use KeyID = 0
- Use KeyID=1 for the current key and KeyID=2 for the new key
- Switch keyID 2 when all mobile devices are notified about the new key (ACK message)

**How to generate GMK?**

- AP chooses a 256-bit cryptographic-quality random number as the GMK
- It is unnecessary to bind the GMK to any identity since group keys are for message protection instead of authentication
Group Temporal Keys

- Group Encryption Key (128 bits)
- Group Integrity Key (128 bits)
- These two keys are concatenated together to form the Group Transient Key (GTK)
- GTK is derived from GMK, a nonce (for liveness) and the MAC address of the AP
- GTK is delivered through a two-way handshake through EAPOL-Key messages

Temporal Key Computation

- All temporal keys should be independent on each other
- PMK, Nonce 1, Nonce 2, MAC 1, and MAC2 are fed into a pseudo Random Generator as the seed to generate random bytes, forming the temporal keys
- Similar for GTK
- Can the same pseudo random generator used for different purposes?
  - Desirable and YES
- RSN and WPA define a set of pseudorandom functions, each incorporating a different text string in to the input, to produce a certain number of bits
  - PRF-128
  - PRF-256
  - PRF-384
  - PRF-512

Pseudorandom Functions

- All the variants of the PRF are implemented using the same algorithm based on HMAC-SHA-1
- Each pseudorandom function takes three parameters and produces the desired number of random bits
  - A secret key
  - A text string identifying the application
  - Some data specific to each case such as nonces. Eg: the starting random number of the nonce counter is PRF-256(Random Number, "Init Counter", MAC||Time)
- PRF-512(PMK, "Pairwise key expansion", MAC1||MAC2||Nonce1||Nonce2)
  - MAC1 is the smallest and Nonce1 is the smallest
- PRF-256(GMK, "Group key expansion", MAC||GNonce)

Nonce Selection

- N-once: A Number used only once with a given key
- When nonces are needed
  - Group keys are refreshed
  - Mobile devices join/leave the network
- Is a calendar clock a good choice?
  - Theoretically YES since a timer never goes back
  - In practical, not practical: Is your clock correct? (synchronization needed for multiple timers)
- A larger nonce counter (256 bits long) initialized with a random number suffices
  - Starting value of the nonce counter = PRF-256(Random Number, "Init Counter", MAC||Network Time (if known))

Summary of Key Establishment

- Authentication Server only knows the PMK
- If authentication is done at the upper layer through an authentication server (e.g., TLS), the procedure authenticates the supplicant and authorizes it to join the network
- If a preshared key is used, authentication is assumed and subsequently verified during the four-way handshake
- Once authorized, the mobile device and access point perform a four-way handshake to generate temporal keys and prove mutual knowledge of the PMK
- The Access point computes and distributes group keys

Summary of Key Hierarchies

- Pairwise Master Key – PMK
  - 256 bits
- Pairwise Transient Key – PTK
  - 512 bits

Protect Key Handshakes

Protect Data
### Summary Of Key Hierarchies

<table>
<thead>
<tr>
<th>Key Type</th>
<th>Length</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group Master Key (GMK)</td>
<td>256 bits</td>
<td>Protect Multicast/Broadcast</td>
</tr>
<tr>
<td>Group Transient Key (GTK)</td>
<td>256 bits</td>
<td>Protect Multicast/Broadcast</td>
</tr>
</tbody>
</table>

- **Data Encryption Key**: 128 bits
- **Data MIC Key**: 128 bits

### What’s Next
- We just talked about the key hierarchies in WPA and RSN.
- Which security cipher to choose?
  - TKIP
  - CCMP