

# ECSE6962: Wireless Ad Hoc Networks

---

## Security

Alhussein Abouzeid  
ECSE, RPI  
November 28th, 2005

# Outline

---

- Security and the protocol stack
- Security approaches and tradeoffs
- Security problems at link and network layers
- Outline of solutions
- Security primitives
- Secure routing protocol example
- Secure packet forwarding
- Remarks on Link-layer security
- Conclusion

# Outline

---

- **Security and the protocol stack**
- Security approaches and tradeoffs
- Security problems at the network and link layers
- Outline of solutions
- Security primitives
- Secure routing protocol example
- Secure packet forwarding
- Remarks on Link-layer security
- Conclusion

# Security and the protocol stack

---

- Early research focused on the challenges posed by ad hoc networks to MAC and routing
- Security becomes a primary concern in a hostile environment
- Additional challenges compared to wired
  - Shared wireless medium
  - Resource constraints (energy and/or bandwidth)
  - Dynamic topology (mobility, users join/leave, etc.)
- The goal is to provide protected multi-hop communication between nodes in a hostile environment
- Security solutions may (should?) span the entire protocol stack
- Solutions for wired networks do not directly apply

# Security solutions span the entire stack

---

Layer	Security issue
Application	Viruses, worms
Transport	Authentication for end-to-end communication
Network	Protecting routing and forwarding protocols
Link	Protecting the wireless MAC protocol; provide link layer security
Physical	Signal jamming DoS attacks

# Outline

---

- Security and the protocol stack
- **Security approaches and tradeoffs**
- Security problems at the network and link layers
- Outline of solutions
- Security primitives
- Secure routing protocol example
- Secure packet forwarding
- Remarks on Link-layer security
- Conclusion

# Security Approaches and tradeoffs

---

## 1. Proactive

- Prevent an attack from happening

## 2. Reactive

- Detect an attack and react quickly
- Effective approaches are likely hybrid
- The key components: prevention, detection, reaction.
- Security typically comes at a cost
  - Computation
  - Communication
  - Management overhead
- Performance may suffer e.g.
  - Scalability
  - Service availability
  - Robustness
- Solutions should consider not only security strength but also performance metrics

# Outline

---

- Security and the protocol stack
- Security approaches and tradeoffs
- **Security problems at the network and link layers**
- Outline of Solutions
- Secure routing protocol example
- Secure packet forwarding
- Remarks on Link-layer security
- Conclusion

# Securing connectivity in ad-hoc networks

---

- “The protection of the network basic functionality to deliver data bits from one node to another.”
- This entails the protection of:
  - Link-layer: one hop connectivity
  - Routing layer: extending connectivity to multiple hops
- Unlike wired, no clear line of defense :
  - the wireless channel is accessible to legitimate as well as malicious attackers
  - each node may function as a router
  - ... so there is no clear point to implement access/traffic control
  - ... the boundary between network “edges” and “core” is blurred
- A malicious attacker can easily assume the router role but intentionally deviate from the protocol specifications

# Network Layer Attacks

---

- Cooperation is assumed but not enforced
- The main network layer operations are
  - routing
  - packet forwarding

These are not separate
- Routing protocols exchange messages and maintain network routing “state” at each node
- Packet forwarding follows accordingly (hopefully!) by intermediate nodes along a path
- thus, we have *routing* and *packet forwarding* attacks

# Routing attacks

---

- Routing attacks: any routing message updates that do not follow the specs – depends on the protocol e.g.
  - e.g. DSR: mess with the RREQ or RREP packets:
    - add, remove, append a node to the list [1]
    - advertise that an active link is broken [1]
  - e.g. AODV: mess with the distance metrics:
    - advertise a smaller metric than the actual distance to the destination;
    - advertise wrong updates with a larger sequence number hence invalidate other valid updates [2]
  - e.g. Wormhole effect [4]: Pair of attackers shortcuts the normal flows by altering the routes
  - Results in: longer routes, non-existent routes, loops, severe congestion on specific nodes/links

# Packet Forwarding Attacks

---

- Drop, modify or duplicate the packets already forwarded
- Inject a large amount of (junk) packets (Denial of Service DoS attack – network layer packet blasting)
  - Cause severe persistent congestion
  - Increased due to MAC contention

# Link Layer Attacks

---

- 802.11 is vulnerable to attacks targeting its channel contention and allocation schemes, e.g.
  - Exploit the binary exponential backoff mechanism to deny access to the channel by its neighbors [10,11]
  - Overhear the NAV field carried in the RTS/CTS which indicates the duration of the reservation, then intentional interfere during the data transmission of a neighbor (also considered a DoS attack)

# Outline

---

- Security and the protocol stack
- Security approaches and tradeoffs
- Security problems at the network and link layers
- **Outline of solutions**
- Security primitives
- Secure routing protocol example
- Secure packet forwarding
- Remarks on Link-layer security
- Conclusion

# Prevention

---

- Mainly achieved by secure routing protocols
- Work in a proactive manner, to prevent the attacker from installing incorrect routing states
- Typical routing protocols + cryptographic primitives to authenticate routing messages (will see examples later)
- From experience, any connected node is not 100% secure. If it's true for wired, it's also true for wireless

# Detection & Reaction

---

- Identification of malicious nodes by noticing their abnormal behavior
- This is intrinsically reactive
  1. End-to-end detection
  2. Local detection through collaborative decision-making
- Reaction: Adjust routing and forwarding operation to avoid or exclude the malicious node

# Network Layer Security

---

- **Secure ad hoc routing protocols**
  - ensure that the routing message exchanged between nodes is consistent with the protocol specification
- **Secure packet forwarding protocols**
  - ensure the packet forwarding behavior of each node is consistent with its routing states

# Outline

---

- Security and the protocol stack
- Security approaches and tradeoffs
- Security problems at the network and link layers
- Outline of solutions
- **Security primitives**
- Secure routing protocol example
- Secure packet forwarding
- Remarks on Link-layer security
- Conclusion

# Message Authentication Primitives

---

- The essential component in any security design is message authentication
- Three crypto primitives for message authentication are:
  1. HMAC
  2. Digital Signature
  3. One-way HMAC

# 1. HMAC

---

- **What it is:**
  - Hash function based Message Authentication Code [15]
    - hash functions are not keyed primitives, ie. do not accommodate naturally the notion of secret key, but we neglect that detail here
  - It was designed to meet the requirements of the IPSEC working group in the IETF, and is now a standard.
  - How can I make sure that the message is actually sent by my partner and it is as sent by the partner (unmodified)?
  - If two nodes share a secret symmetric key  $K$ , they can generate and verify a message authenticator  $h_K(.)$  using a cryptographic one way hash function  $h$ .

# HMAC (Cont'd)

---

- **How it works:**
  - when party A transmits a message to party B, it appends to the message a value called the authentication tag, computed by the MAC algorithm as a function of the transmitted information and the shared secret key.
  - At reception, B recomputes the authentication tag on the received message using the same mechanism (and key) and checks that the value he obtains equals the tag attached to the received message.
  - Only if the values match is the information received considered as not altered on the way from A to B.
  - (In cryptography, the goal is to prevent forgery, namely, the computation, by the adversary, of a message (not sent by the legitimate parties) and its corresponding valid authentication tag.)

# (HMAC cont'd)

---

- **Advantages:**
  - The computation is very efficient (good for handhelds)
- **Disadvantages:**
  - Establishing a secret key in the first place is not trivial
  - Does not scale well: a network of  $n$  nodes requires  $n(n-1)/2$  keys to be established (and maintained).
  - Does not suit broadcast
- e.g. SRP for DSR [13]

# 2. Digital Signatures

---

- **Asymmetric Key Cryptography**
  - Digital signatures use asymmetric key crypto, so here is an intro...
  - key material is bound to a single user
    - a private key, to which only the user has access, and
    - a public key, which may be published or distributed on request.
  - Each key generates a corresponding transformation function
  - The functions are inversely related, i.e., if one function is used to encrypt a message, the other is used to decrypt the message
  - The advantage of a public-key system is that two users can communicate securely without exchanging secret keys.
  - The originator encrypts the message using the recipient's public key.
  - Only the recipient's private key can be used to decrypt the message.
  - This is due to the computational infeasibility of inverting the public key transformation function. In other words, without the recipient's private key, it is computationally infeasible for the interceptor to transform the ciphertext into its original plaintext.

# 2. Digital Signatures

---

- **What it is:**

- Based on asymmetric key (also called public-key) cryptography e.g. RSA (see next slide)
- A digital signature is a cryptographic checksum computed as a function of a message and a user's private key.
- A digital signature is different from a hand-written signature, in that hand-written signatures are constant, regardless of the document being signed. A user's digital signature varies with the data. For example, if a user signs five different messages, five different signatures are generated. Each signature, however, can be authenticated for the signing user.
- Due to the efficiency drawbacks of public-key cryptography, a user often signs a condensed version of a message, called a message digest, rather than the message itself. Message digests are generated by (1-way & collision-free) hash functions.

---

- So, digital signature goes as follows:

At originator:

- Apply hash function to message to generate message digest
- Encrypt message digest using own private key → signature
- Append signature to the message
- Send message+signature.

At intended receiver:

- Apply hash function to message to generate message digest\_B
- Decrypt signature using originator public key → message digest\_A

Note: this does not protect the message from being seen by others.  
But it proves to receiver that the message is authentic and unaltered.

---

- So, (secure) digital signature goes as follows:

At originator:

- Apply hash function to message to generate message digest
- Encrypt message digest using own private key → signature
- Append signature to the message
- (Encrypt all of this using receiver's public key)
- Send encrypted(message+signature)

At intended receiver:

- (Decrypt using own private key → originator signature + message)
- Apply hash function to message to generate message digest\_B
- Decrypt signature using originator public key → message digest\_A

# Advantages

---

- **Scales better**
  - a digital signature (but not secure message) can be verified by any node given that it knows the public key of the signing node,
  - hence is scalable to a large numbers of receivers – only a total number of  $n$  public/private key pairs

---

## ▪ Disadvantages

- Asymmetric key involves much more computation overhead in signing/decrypting and verifying/encrypting operations.
  - Hence vulnerable to a DoS attack where an attacker may feed a victim node with a large number of bogus signatures to exhaust the victim's computation resources for verifying them.
  - Each node also needs to keep a certificate revocation list (CRL) of revoked certificates.
  - Also, an attacker may fool a receiver by impersonating someone else (say I post I am Bob and my public key)– solution by “certificate” requires centralized authority
- SAODV [2] and ARAN [3] take the digital signature approach.

# 3. One-way HMAC Key Chain

---

## How it works:

- To generate a key chain of length  $n+1$ , the first element of the chain  $h_0$  is randomly picked and then the chain is generated by repeatedly applying a one-way function  $H$
- A one-way hash function maps an input to an output of a fixed length, which is the new key
- The keys are then used and revealed in the reverse order, starting from the last generated key  $h_n$
- Any key  $h_j$  can be verified from  $h_i$  ( $0 \leq i < j \leq n$ ) to be indeed an element in the chain by repeatedly applying  $H$  for  $j - i$  times
- **Advantages:**
  - Less computation than asymmetric crypto
  - One authenticator can be verified by a large number of receivers

---

## Disadvantages:

- Requires synchronization
  - Receivers need to buffer messages for verification when the key is revealed – this delay may reduce routing protocol responsiveness to topology changes
  - The release of the key requires a second round of communication
  - Storage of the hash chain limits scalability
- 
- e.g. TESLA, SEAD (for DSDV) and Ariadne (for DSR).

# TESLA (example of one way HMAC key chain)

---

- Unicast authentication is cheap, using symmetric key authentication, but doesn't scale for broadcast.
- Secure broadcast authentication requires an asymmetric primitive
- TESLA differs from asymmetric protocols (e.g. RSA) in that it achieves this asymmetry from clock synchronization and delayed key disclosure, rather than from computationally expensive one-way functions

# TESLA (Cont'd)

---

- *Timed Efficient Stream Loss-tolerant Authentication* [8]
- Requires time synchronization and causes delay in authentication
- Idea:
  - The sender commits to a random key  $k$  without revealing it and transmits it to the receivers.
  - The sender then attaches a message authenticating code to the next packet  $P_i$  and uses the key  $k$  as the MAC key.
  - In a later packet  $P_{i+1}$ , the sender decommits to  $k$  (i.e. it discloses the key), which allows the receivers to verify the commitment and the MAC of packet  $P_i$ .
  - If both verifications are correct, and if it is guaranteed that packet  $P_{i+1}$  was not sent before packet  $P_i$  was received, then a receiver knows that the packet  $P_i$  is authentic.
  - To start this scheme, the sender uses a regular signature scheme to sign the initial commitment. All subsequent packets are authenticated through chaining.
  - **Security condition:** A data packet  $P_i$  arrived *safely*, if the receiver can unambiguously decide, based on its synchronized time and network delay, that the sender did not yet send out the corresponding key disclosure packet.
- **Packet loss tolerance**
  - Sender precomputes a sequence of  $n$  key values, called key chain (all one way i.e. each one is not invertible)
  - If receiver receives packet  $P_i$ , any subsequent packet reception allows it to compute  $k_i$  and  $k_i'$  and verify the authenticity of  $P_i$

# TESLA example [9]

---

$$P_{i-1} = \langle D_{i-1}, \text{MAC}(K'_{i-1}, D_{i-1}) \rangle$$

$$D_{i-1} = \langle M_{i-1}, F(K_i), K_{i-2} \rangle$$

pick a fresh key  $K_{i+1}$

$$P_i = \langle D_i, \text{MAC}(K'_i, D_i) \rangle$$

$$D_i = \langle M_i, F(K_{i+1}), K_{i-1} \rangle$$

pick a fresh key  $K_{i+2}$

$$P_{i+1} = \langle D_{i+1}, \text{MAC}(K'_{i+1}, D_{i+1}) \rangle$$

$$D_{i+1} = \langle M_{i+1}, F(K_{i+2}), K_i \rangle$$

where

$K'_i = F'(K_i)$ ;  $F$  and  $F'$  are two hash functions

- The example shows the transmission of three consecutive packets
- When the  $i+1$  packets is received, the receiver can:
  - Verify that  $K_i$  is correct by generating  $F(K_i)$  and comparing it with the one received in packet  $i-1$
  - Compute  $F'(K_i)$  and generate MAC of  $P_i$ , then compare it with the one received in packet  $i$ .
  - The above also authenticate  $F(K_{i+1})$
- The only way to break this security chain is for an attacker to get  $P_{i+1}$  before the receiver gets  $P_i$  (hence it would know  $k_i$  and take over the sequence without the receiver being able to detect that.)

---

$$F^j(x) = F^{j-1}(x)$$

$$K_0 = F^n(K_n)$$

$$K_i = F^{n-i}(K_n)$$

for example:

$$K_0 = F^3(K_3)$$

$$K_1 = F^2(K_3)$$

$$K_2 = F(K_3)$$

$$K_3 = K_3$$

- What happens if a packet in the sequence is lost?: make a key chain
- Given  $K_j$ , an attacker cannot generate any  $K_j$ ,  $j > i$ .
- If a receiver receives packet  $P_i$ , any subsequent packet will allow it to compute  $K_i$  and  $K'_i$  as well as verify the authenticity of  $P_i$ .
- For example, suppose, in a sequence of packets,  $F^3(K_3)$  is received,  $F^2(K_3)$  is lost and  $F(K_3)$  is received. Then receiver can run  $F(F(K_3))$  to get  $F^2(K_3)$  so as to validate the received packet.

# Outline

---

- Security and the protocol stack
- Security approaches and tradeoffs
- Security problems at the network and link layers
- Outline of Solutions
- **Secure routing protocol example**
- Secure packet forwarding
- Remarks on Link-layer security
- Conclusion

# Secure Source Routing: Ariadne [1]

---

- Ariadne is designed for source routing protocols
- $S$  performs a Route Discovery for target  $D$ , they share the secret keys  $K_{SD}$  and  $K_{DS}$ , respectively, in each direction.
- Target authenticates RREQ
  - To convince the target of the legitimacy of each field in a ROUTE REQUEST, the initiator includes a MAC computed with key  $K_{SD}$  over unique data, e.g. timestamp. The target can easily verify the authenticity.

# Node list authentication in Ariadne

---

- **Data authentication:**
  - initiator wants to authenticate each individual node in the node list of the RREP
  - target wants to authenticate each node in the node list of the ROUTE REQUEST
- **Ariadne proposes three techniques for node list authentication based :**
  - **Standard MACs**
    - is most efficient, but requires pairwise shared keys between all nodes.
    - the MAC list in the REQUEST is computed using a key shared between the target and the current node.
  - **Digital signatures**
    - the REQUEST includes a signature list
  - **TESLA**
    - target buffers the REPLY until intermediate nodes can release the corresponding TESLA keys
- **Protection against node removal**
  - use one-way hash functions to verify that no hop was omitted -- called *per-hop hashing*.

# Outline

---

- Security and the protocol stack
- Security approaches and tradeoffs
- Security problems at the network and link layers
- Outline of Solutions
- Secure routing protocol example
- **Secure packet forwarding**
- Remarks on Link-layer security
- Conclusion

# Secure packet forwarding

---

- It is possible for a malicious node to correctly participate in the route discovery phase but fail to correctly forward data packets.
- The security solution should ensure that each node indeed forwards packets according to its routing table.
- This is typically achieved by the reactive approach because attacks on packet forwarding
- cannot be completely prevented
- Two key components of this solution are
  - a detection technique and
  - a reaction scheme

# Detection

---

- Each node may perform localized detection by overhearing ongoing transmission
  - Limited by channel errors, interference, mobility
  - Need mechanisms to avoid false accusations (slander)
  - e.g.
    - watchdog solution[10] for DSR
      - A sends to B and knows (from header) that B's next hop is C
      - Assuming symmetric links, A will check to see if it overhears B's transmission to C, if not it increases a counter
      - If the counter passes a threshold it reports B's behavior to the source
    - Solution[11] works for AODV
      - Same concept after adding a “next\_hop” field in AODV
      - Also considered other kinds of attacks
    - Solution [9] utilizes an Ack-based scheme
      - Used on demand when a path loss rate is suspicious
      - Sends encrypted probes

# Reaction

---

- Actions triggered to protect the network from future attacks launched by the detected malicious node
- Usually related to the prevention component
  - Certificate revoked
  - Chosen as a router with smaller probability
- Two categories
  - Global reaction
    - All nodes react in the same way (e.g. exclude the malicious node)
    - E.g. [11] multiple nodes who reached a consensus revoke the certificate of the malicious node hence isolating it
  - End-host reaction
    - Each node makes its own decision
    - E.g. *pathrater* [10] allows each node to keep its own rating for the other nodes it knows about

# Outline

---

- Security and the protocol stack
- Security approaches and tradeoffs
- Security problems at the network and link layers
- Outline of Solutions
- Secure routing protocol example
- Secure packet forwarding
- **Remarks on Link-layer security**
- Conclusion

# 802.11 MAC

---

- The attacker may exploit its binary exponential backoff scheme to launch DoS attacks [5, 6].
  - Reference [5] uses simulations to show that implementing a fair MAC protocol is a necessary but insufficient technique to solve the problem.
  - Need a MAC with fairness guarantees
  - [6] proposes a MAC protocol that seeks to detect and react to MAC-layer misbehaviors (i.e. it is reactive)
    - The backoff timer is provided by the receiver (hence receiver can detect misbehavior)
  - NAV field in RTS/CTS is vulnerable to DoS attacks [12]
    - Attacker may only need to interfere on a few bits – power consumption favors the adversary side
- ...Not clear how to prevent such resource consumption-based DoS attacks

# IEEE 802.11 WEP

---

- WEP [14] (Wired Equivalent Privacy)
  - is 802.11's optional encryption standard implemented in the MAC layer
- WEP has several problems regarding the vulnerability of the cryptographic techniques to attacks
- 802.11i WPA [13] seems to have fixed these weaknesses

# Conclusion

---

- Touched on a few issues regarding securing ad hoc networks
- Research on adhoc networks security is still in its early stage
- The existing proposals are typically attack-centric:
  - first identify several security threats
  - and then enhance the existing protocol or propose a new protocol
- This may not work well under unanticipated attacks
  - A more ambitious goal is to embed security measures that can protect against unanticipated attacks – a broader space e.g. treat malfunction as an attack?
- Need better tools to understand tradeoffs between security strength, communication overhead, computation complexity, energy consumption, and scalability (largely unexplored)

# References

---

- [1] Y. Hu, A. Perrig, and D. Johnson, “Ariadne: A Secure On-demand Routing Protocol for Ad Hoc Networks,” *ACM MOBICOM*, 2002.
- [2] M. Zapata, and N. Asokan, “Securing Ad Hoc Routing Protocols,” *ACM WiSe*, 2002.
- [3] B. Dahill *et al.*, “A Secure Protocol for Ad Hoc Networks,” *IEEE ICNP*, 2002.
- [4] Y. Hu, A. Perrig, and D. Johnson, “Packet Leashes: A Defense against Wormhole Attacks in Wireless Networks,” *IEEE INFOCOM*, 2002.
- [5] V. Gupta, S. Krishnamurthy, and M. Faloutsos, “Denial of Service Attacks at the MAC Layer in Wireless Ad Hoc Networks,” *IEEE MILCOM*, 2002.
- [6] P. Kyasanur, and N. Vaidya, “Detection and Handling of MAC Layer Misbehavior in Wireless Networks,” *DCC*, 2003.
- [7] P. Papadimitratos, and Z. Haas, “Secure Routing for Mobile Ad Hoc Networks,” *CNDS*, 2002.
- [8] Adrian Perrig, Ran Canetti, J.D. Tygar, and Dawn Song. Efficient Authentication and Signing of Multicast Streams over Lossy Channels. In *IEEE Symposium on Security and Privacy*, pages 56–73, May 2000.
- [9] B. Awerbuch *et al.*, “An On-Demand Secure Routing Protocol Resilient to Byzantine Failures,” *ACM WiSe*, 2002.
- [10] S. Marti *et al.*, “Mitigating Routing Misbehavior in Mobile Ad Hoc Networks,” *ACM MOBICOM*, 2000.
- [11] H. Yang, X. Meng, and S. Lu, “Self-Organized Network Layer Security in Mobile Ad Hoc Networks,” *ACM WiSe*, 2002.
- [12] G. Noubir and G. Lin, “Low-Power DoS Attacks in Data Wireless LANs and Countermeasures,” *ACM MobiHoc*, Poster Session, 2003.
- [13] IEEE Std. 802.11i/D30, “Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Specification for Enhanced Security,” 2002.
- [14] IEEE Std. 802.11, “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” 1997.
- [15] M. Bellare, R. Canetti, H. Krawczyk, “Keying Hash Functions for Message Authentication,” *Proceedings of Advances in Cryptology - CRYPTO '96: 16th Annual International Cryptology Conference*, Santa Barbara, California, USA, August 1996.
- [17] H Yang, H Luo, F Ye, S Lu, L Zhang, “Security in mobile ad hoc networks: Challenges and solutions,” *IEEE Wireless Communications Magazine*, February 2004.