Secure, Efficient Key Management for False Data Detection in Wireless Visual Sensor Networks using Dynamic Key Chaining

Gowun Jeong

Advanced Information Security, CS, KAIST

May 13, 2010

Outline

1. Motivation
2. Assumptions
   - Network Topology
   - Intrusion Detecting Process and Attack Scenarios
3. False Data Detection
   - Message Authentication Code (MAC)
   - Dynamic Key Chaining
   - False Data Detection and Recovery Protocol (FDDR)
4. Performance Analysis
5. Limitations
6. Conclusion

Outline

1. Motivation
2. Assumptions
   - Network Topology
   - Intrusion Detecting Process and Attack Scenarios
3. False Data Detection
   - Message Authentication Code (MAC)
   - Dynamic Key Chaining
   - False Data Detection and Recovery Protocol (FDDR)
4. Performance Analysis
5. Limitations
6. Conclusion

Motivation

Secure, Efficient Wireless Visual Sensor Networks (WVSNs)

- Terminology
  - **wireless** limited power in computation, memory and energy
  - **VSNs** a large amount of data to handle and transfer for intrusion detection
- Objectives
  - **secure** to be robust against false data injection (FDI) by T compromised wireless nodes
    - Origin and data integrity
    - How to detect and if FDI actually occurs, then how to recover?
  - **efficient** to reduce additional energy consumption and computation, memory and communication overhead

Outline

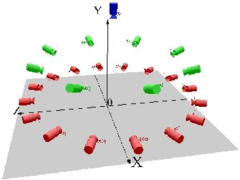
1. Motivation
2. Assumptions
   - Network Topology
   - Intrusion Detecting Process and Attack Scenarios
3. False Data Detection
   - Message Authentication Code (MAC)
   - Dynamic Key Chaining
   - False Data Detection and Recovery Protocol (FDDR)
4. Performance Analysis
5. Limitations
6. Conclusion

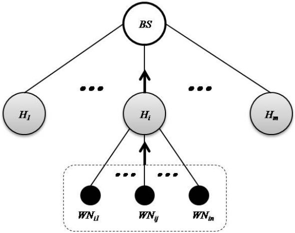
Outline

1. Motivation
2. Assumptions
   - Network Topology
   - Intrusion Detecting Process and Attack Scenarios
3. False Data Detection
   - Message Authentication Code (MAC)
   - Dynamic Key Chaining
   - False Data Detection and Recovery Protocol (FDDR)
4. Performance Analysis
5. Limitations
6. Conclusion

Network Topology

- Cluster-based densely deployed network
- One-hope communication allowed
- *BS* (Base Station) and *H*s (Heads) wired and strongly trusted
- *WN*s (Wireless Nodes) for sensing in rotation for energy efficiency
- *H*s for sensing and data aggregating
- *BS* for data aggregating





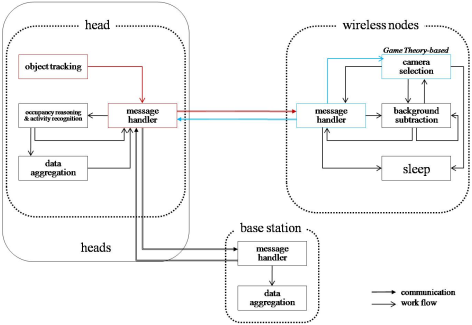
Outline

1. Motivation
2. Assumptions
   - Network Topology
   - Intrusion Detecting Process and Attack Scenarios
3. False Data Detection
   - Message Authentication Code (MAC)
   - Dynamic Key Chaining
   - False Data Detection and Recovery Protocol (FDDR)
4. Performance Analysis
5. Limitations
6. Conclusion

Vulnerability of WVSNs

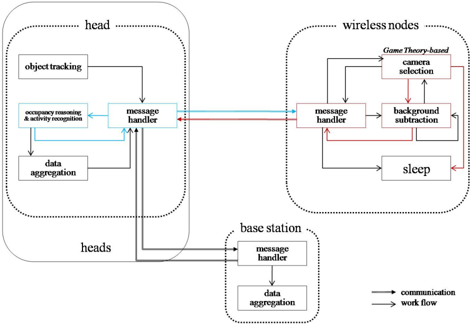
- Types of attacks
  - Physical node capturing: Attackers could steal cryptographic keys
  - Communication channel attacks: Attackers could know some frequently used channels
  - Sybil attacks: Attackers could pretend legitimate nodes
- Potential attacks
  - Since *BS* and *H*s never compromise, only channel attacks occur during broadcasting from them
  - All the attacks could take place during reporting from *WN*s

Case 1: Channel Attacks



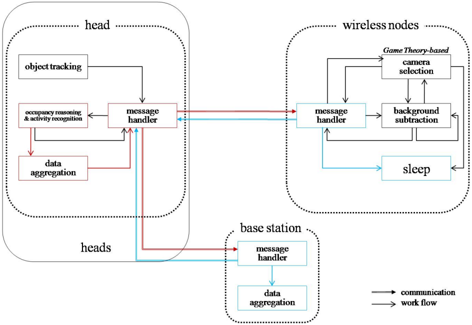
1. Before any intrusion occurs, H_i's all neighbours WN_{ij}s sleep
2. Once H_i perceives an abnormal event, it broadcasts that to all WN_{ij}s by sending message **activate** while stopping its sensing
3. Each WN_{ij} checks the message and selects a minimal set of cameras that should be turned on; if the verification is unsuccessful, it recommends H_i to use another frequency because the current one is unsecured

Case 2: Node Capturing and Sybil Attacks



1. Each WN_{ij} sends the following three types of messages in situations
   - after *camera selection*, reports **the IDs of selected cameras**
   - after *background subtraction*, sends **the resulted image**
   - after seen no objects to observe, forward message **no objects**
2. H_i do a semantic check on the message; if the verification is unsuccessful, H_i drops the message and announces it to all WN_{ij}s since the key being used is disclosed

Case 3: Channel Attacks



1. After receiving **no objects** from all selected WN_{ij}s, H_i broadcasts them to sleep, and then reports the lastest activity recognition result to *BS*
2. Same as in Case 1, against undesirable verification result, WN_{ij} warns H_i to use another frequency for transmission
3. After checking the message's integrity, *BS* broadcasts all H_is the received message; if the verification is unsuccessful, *BS* requires H_i to use another frequency

Outline

1. Motivation
2. Assumptions
   - Network Topology
   - Intrusion Detecting Process and Attack Scenarios
3. False Data Detection
   - Message Authentication Code (MAC)
   - Dynamic Key Chaining
   - False Data Detection and Recovery Protocol (FDDR)
4. Performance Analysis
5. Limitations
6. Conclusion

Outline

1 Motivation

2 Assumptions
- Network Topology
- Intrusion Detecting Process and Attack Scenarios

3 False Data Detection
- Message Authentication Code (MAC)
- Dynamic Key Chaining
- False Data Detection and Recovery Protocol (FDDR)

4 Performance Analysis

5 Limitations

6 Conclusion

Message Authentication Code (MAC)

- Origin and data integrity using MAC
  - Generally, to check message integrity, a MAC generating, hash function which is easy to compute, but hard to reverse is used
  - If a sender sends M and $MAC_K(M)$ using shared key K with its receiver, the receiver verifies if the received M produces the same MAC as the received $MAC_K(M)$
  - Most of channel attacks can be detected by using MAC
- Limitations on using mere MAC
  - What if collisions occur?
    - Collision: $MAC_K(M) = MAC_K(M')$ for $M \neq M'$
    - There could be more than 15,000 packets to deliver per image assuming a packet includes 32 bytes
    - $|MAC| = 4\ bytes$, $P(two\ match) = 1 - e^{(-7{,}500 \times 14{,}999/2^{32})} \approx 0.0259$
  - What if a key used to produce the MAC value is disclosed?
    - An attackers can replace message M with message M' by sending M' and $MAC_K(M')$ with disclosed key K

Outline

1. Motivation
2. Assumptions
   - Network Topology
   - Intrusion Detecting Process and Attack Scenarios
3. False Data Detection
   - Message Authentication Code (MAC)
   - Dynamic Key Chaining
   - False Data Detection and Recovery Protocol (FDDR)
4. Performance Analysis
5. Limitations
6. Conclusion

Assumptions

- Every node has the same encryption/decryption (symmetric) function, MAC function and one-way function to generate a key chain
- Only pairwise key is used to communicate and all they are pre-distributed before communication
- Saying K_{ij} key between H_i and WN_{ij}, H_i securely requires every WN_{ij} to produce a key chain of certain length l using the one-way function by $f_{K_{ij}}(KC_{t+1}) = KC_t$ before starting to communicate[1]

[1] SPINS: Security Protocols for Sensor Networks, Perring et al., *Wireless Networks*, (8)521-534, 2002

Key Management: Dynamic Key Chaining

- Commitment KC_c: the last key of the current key chain
- A sender (either H_i or WN_{ij}) computes MAC for time interval t using the commitment, and sends M and $MAC_{K_t}(M)$ by employing session key $K_t = K_{ij} \oplus KC_c$ to its receiver while erasing KC_c from the key chain for the receiver
- The receiver verifies and decrypts the message, and then erase KC_c from the key chain, too
- When there is no commitment left, they compute another key chain setting that K_{ij} is the last commitment
- When a key (KC_c, K_{ij} or K_t) is stolen, they generate another key chain setting that the next commitment is the seed K_{ij} for the chain

Formal Results

Lemma 1

This dynamic key chaining guarantees no collision with high probability since a key to produce MAC is employed only once in each delivery.

- Each time to generate a key chain a different key is taken as the seed, and thereby, the resulted keys are different from those previously produced with high probability

Formal Result

Lemma 2

This dynamic key chaining is resilient against any size of compromised node set with high probability, while requiring every wireless node to store $O(|K|(l+2))$ keys, to transit only $|MAC|$ additional bytes and to do $O(|MAC|(\alpha+\beta)+|K|(l+1)(\frac{\alpha}{l+1}+\beta))$ more computation for the number of legitimate data packets α and the number of false data packets β.

- Since only individual, pairwise communication is allowed, any size of compromised set hardly discloses others' secure information
- Setting the key size is big enough as $|K| = 32$ *bytes*, the complexity of breaking a key is $\Omega(2^{39})$[2]
- Only known either the current commitment or the shared key, the session key is not revealed easily
- Once any key being used is disclosed, a new, intractable key chain is computed by the one-way function since the seed is hardly obtained

[2] Key Recovery Attacks of Practical Complexity on AES Variants with up to 10 Rounds, Biryukov et al., *ePrint Archive*, 2010

Outline

1. Motivation
2. Assumptions
   - Network Topology
   - Intrusion Detecting Process and Attack Scenarios
3. False Data Detection
   - Message Authentication Code (MAC)
   - Dynamic Key Chaining
   - False Data Detection and Recovery Protocol (FDDR)
4. Performance Analysis
5. Limitations
6. Conclusion

FDDR against Cases 1 and 3

1.	$S \Rightarrow R_i$s	$E_{K_t}(D)$ and $MAC_{K_t}(E_{K_t}(D))$
2.	R_i	verify if the receive message generates $MAC_{K_t}(E_{K_t}(D))$
3.	$R_i \rightarrow S$	[verified] success [unverified] request to resend the same message using another frequency

- S: BS or H_i
- Rs: Hs or WN_{ij}s
- E: the encryption function
- Since the senders are strongly trusted, only possible FDI is sending $E_{K_t}(D) + FD$ for false data FD; so, this is easily detected by computing MAC

FDDR against Case 2

1.	$WN_{ij} \rightarrow H_i$	$E_{K_t}(D_{ij})$ and $MAC_{K_t}(E_{K_t}(D_{ij}))$
2.	H_i	integrity check and **semantic check**
3.	$H_i \Rightarrow WN_{ij}$s	[verified] success [unverified integrity] request to resend using another frequency [unverified semantic] failure (and request to reselect cameras)

- FDI can take any form of $\{E_{K_t}(D) + FD$, $E_{K_t}(D + FD)$ and $MAC_{K_t}(E_{K_t}(D + FD))$, $E_{K_t}(FD)$ and $MAC_{K_t}(E_{K_t}(FD))\}$
- The first is detected by integrity check computing MAC
- The rest is verified by semantic check according to the types of message as follows
  - IDs of selected cameras/no objects: FDI occurs if $\bigwedge_j D_{ij} \neq D_{ij}$
  - resulted image: FDI occurs by node WN_{ik} if $\forall j \exists k[(act(D_{ik}) = \sim act(D_{ij}) \wedge \sim act(D_i)) \wedge (act(D_i) \rightarrow act(D_{ij}))]$ for the activity recogniser act; then, such D_{ik} is discarded

Formal Result

Lemma 3

A false data packet injected by any compromised node can be detected in one hop communication.

- The verification process occurs every hop
- Against the messages, such as IDs of cameras and no objects, the semantic check perfectly works
- Against the image messages, the semantic check largely relies on the performance of the activity recogniser; if it produces highly accurate recognition, the lemma can be achieved

Outline

1. Motivation
2. Assumptions
   - Network Topology
   - Intrusion Detecting Process and Attack Scenarios
3. False Data Detection
   - Message Authentication Code (MAC)
   - Dynamic Key Chaining
   - False Data Detection and Recovery Protocol (FDDR)
4. Performance Analysis
5. Limitations
6. Conclusion

Performance Analysis

- Cost comparison with the following three existing studies
  - TRAD: a traditional message authentication scheme;
  - SPINS: SPINS[1] using a key chain in order for broadcasting; and
  - DAA: Data Aggregation and Authentication Protocol[3] employing two different MACs
- in three different measures
  - Memory overhead;
  - Computation overhead; and
  - Communication overhead

[1] SPINS: Security Protocols for Sensor Networks, Perring et al., *Wireless Networks*, (8)521-534, 2002

[3] Integration of False Data Detection with Data Aggregation and Confidential Transmission in Wireless Sensor Networks, Ozdemir and Cam, *IEEE/ACM Transactions on Networking*, 2009

Cost Comparison in Memory Overhead

Size	TRAD	SPINS	DAA	FDDR
MAC	1	1	$4(T+1)$	1
key	$3\leq$	$3\leq$	$2T\geq$	$l+2\geq$

- Other than DAA, all employ only one MAC to authenticate a packet
- DAA requires $2(T+1)$ MACs for one packet in a pair
- TRAD and SPINS allow that a wireless node directly communicates its neighbours, its head and even the base station using pairwise keys, a group key and a key shared with the base station
- In DAA, an aggregator should store every key shared with its T neighbours and T monitors
- In FDDR, a wireless node is allowed to communicate only with its head

Cost Comparison in Computation Overhead

Computation	TRAD	SPINS	DAA	FDDR
MAC	1	1	$4(T+1)$	1
Aggregation	0	0	$T+1$	2
Encryption/ Decryption	2	2	$T+2$	2
Key Generation	0	0	0	$l+1$

- MAC computation has been already discussed before
- To avoid forwarding redundant information, data aggregation is necessarily required; however, only DAA and FDDR where it is achieved in a head and the base station does
- In DAA, encryption/decryption is carried out in every monitor for one packet as well
- Only FDDR dynamically generates keys

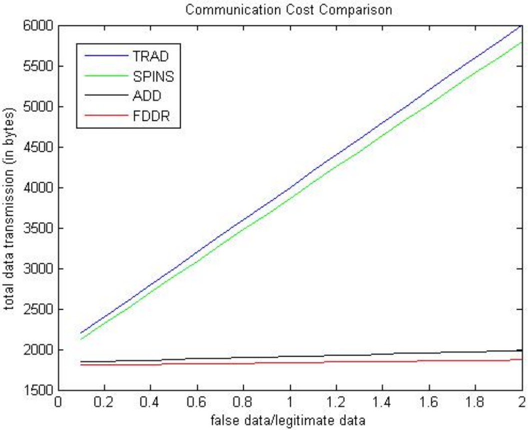
Cost Comparison in Communication Overhead

D_{TRAD}	the amount (in bytes) of data transmission using TRAD of a 8-byte MAC
D_{SPINS}	the amount (in bytes) of data transmission using SPINS of a 6-byte MAC
D_{ADD}	the amount (in bytes) of data transmission using ADD of two 4-byte MACs
D_{FDDR}	the amount (in bytes) of data transmission using FDDR of a 4-byte MAC
L_{tos}	the length (in bytes) of an authenticated and encrypted data packet
α	the number of data packets generated by legitimate nodes
β	the number of false data packets injected by up to T compromised nodes
H_d	the average number of hops between two consecutive data aggregators
H	the average number of hops that a data packet travels in the network
K	the size (in bytes) of key from the key chain
γ	the average number of keys travelled in the network

$$
\begin{aligned}
D_{TRAD} &= (L_{tos} + 8)H(\alpha + \beta) \\
D_{SPINS} &= ((L_{tos} + 6)H + \gamma K)(\alpha + \beta) \\
D_{ADD} &= (L_{tos} + 4)(\alpha H + \beta H_d) + T(L_{tos} + 4)(\alpha + \beta) + \frac{4T}{T+1}(\alpha + \beta) \\
D_{FDDR} &= (L_{tos} + 4)(\alpha H + \beta)
\end{aligned}
$$

Simulation Result for Communication Cost Comparison

- $L_{tos} = 32$, $H = 50$, $\underline{H_d = 1}$, $\underline{\gamma = 1}$, $K = 32$, $\underline{T = 1}$ and $0.2 \leq \beta/\alpha \leq 2$



Outline

1. Motivation
2. Assumptions
   - Network Topology
   - Intrusion Detecting Process and Attack Scenarios
3. False Data Detection
   - Message Authentication Code (MAC)
   - Dynamic Key Chaining
   - False Data Detection and Recovery Protocol (FDDR)
4. Performance Analysis
5. **Limitations**
6. Conclusion

Limitations

- Given a limited bandwidth available to a wireless node, FDDR might not recover from a number of channel attacks
- During repeating camera selection, FDDR could result in losing some invaluable data of the occurring event
- There needs to adjust the length of key chain, appropriately, considering tradeoff between memory and computation overhead
- FDDR deals only with the network that contains **a non-negligible portion of wired nodes**
  - At least one node in an area should do much more computation, such as occupancy reasoning and activity recognition based on multiple images
  - At least one node should constantly monitor its area of responsibility for accurate object tracking
- Once the current commitment is revealed, its previous keys are also disclosed (even though deleting all previous keys used so far)

Outline

1 Motivation

2 Assumptions
- Network Topology
- Intrusion Detecting Process and Attack Scenarios

3 False Data Detection
- Message Authentication Code (MAC)
- Dynamic Key Chaining
- False Data Detection and Recovery Protocol (FDDR)

4 Performance Analysis

5 Limitations

6 Conclusion

Conclusion

- No collision occurs even though the network have to handle a large pool of data packets
- For additional, but reasonable memory consumption to store a key chain, higher security is guaranteed with high probability
- Adjusting the length of key chain, relatively less memory, less computation and less transmission overhead can be assured