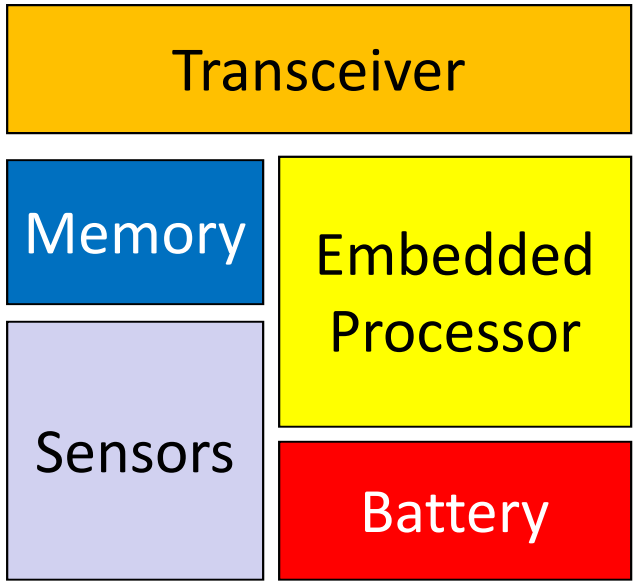
A Forward & Backward Secure Key Management in Wireless Sensor Networks for PCS/SCADA

Hani Alzaid, DongGook Park, Juan Gonzalez, Colin Boyd, and Ernest Foo

Key Management in WSNs for PCS/SCADA

- Introduction.
  - WSNs & SCADA.
- Adversary Model.
- Related Work.
- The proposed key management.
  - Group key update.
  - Pairwise key update.
  - Delivery Failure Management
- Conclusion

Introduction: WSNs



Limited Storage

1Kbps - 1Mbps,
3-100 Meters,

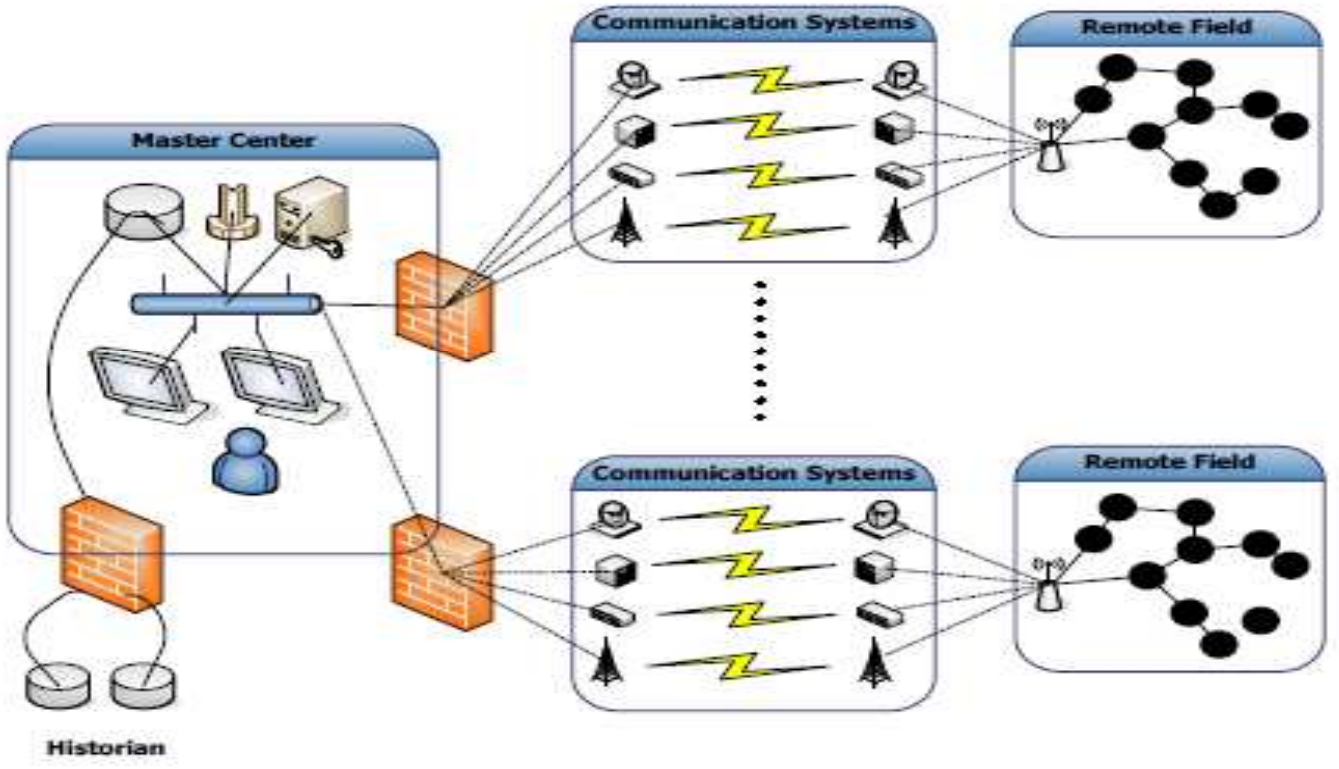
Slow Computations

Limited Lifetime

Introduction: SCADA

- The 3rd generation is a combination of legacy and modern technology.
- It has become an open system architecture rather than a vendor controlled architecture as in the 2nd generation.
- It uses open standards and protocols which facilitate functionalities distribution of SCADA.

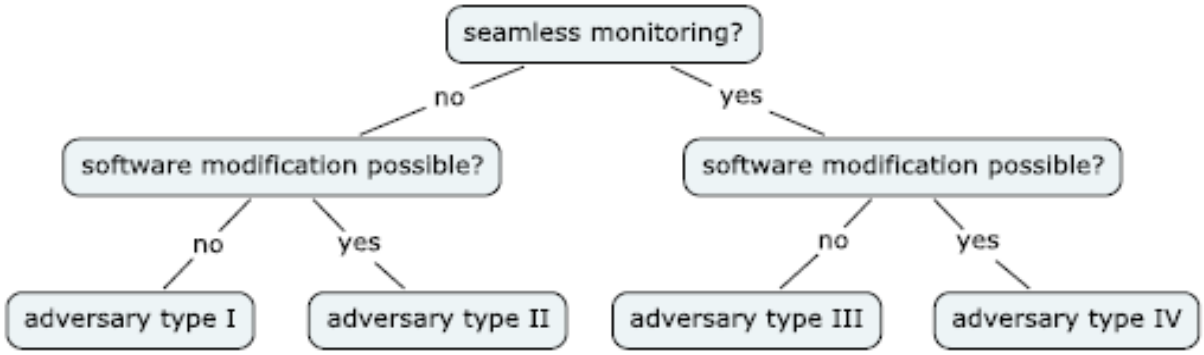
Introduction: SCADA



Adversary Model

- The most challenging threat in WSNs is the node compromise.
- The adversary can compromise
  - All the credentials stored in sensors.
  - All the software codes installed within the sensors, especially random number generation functions.
- It cannot compromise the network manager.

Adversary Model



- The considered adversary can be classified according to the following criteria:
  - Read and modify all the S/W codes.
  - Carry out seamless monitoring of all the subsequent key update protocol.

Related Work

- Several papers dealing with key management designs for SCADA.
  - They used heavy cryptographic mechanism.
  - Do not consider the integration of WSNs with SCADA.
- The only work that considers the integeration, proposed by Nilsson et al. (**Key management and secure software updates in wireless process control environments).**

Related Work

- Nilsson et al. designed two key update protocols:
  - The 1st protocol updates the pairwise symmetric key between M and N.
  - The 2nd protocol updates the global or group key among M and G.
- They claimed that these protocols provide both forward and backward secrecy (past and future key secrecy). **It is not the case!**

Nilsson et al.'s work

The Group Key Update

M: generates a new group key $K_G^{`}$ and a random r_M

$$1.M \rightarrow N:\{K_G^{'}, r_M\}_{K_{MN}}$$

$$2.M \leftarrow N:MAC_{K_G^{'}}(N, r_M)$$

- The whole value of the new group key are directly carried by the protocol messages, encrypted under the pairwise key K_{MN}.

Nilsson et al.'s work

The Pairwise Key Update

N: generates a random number r_N

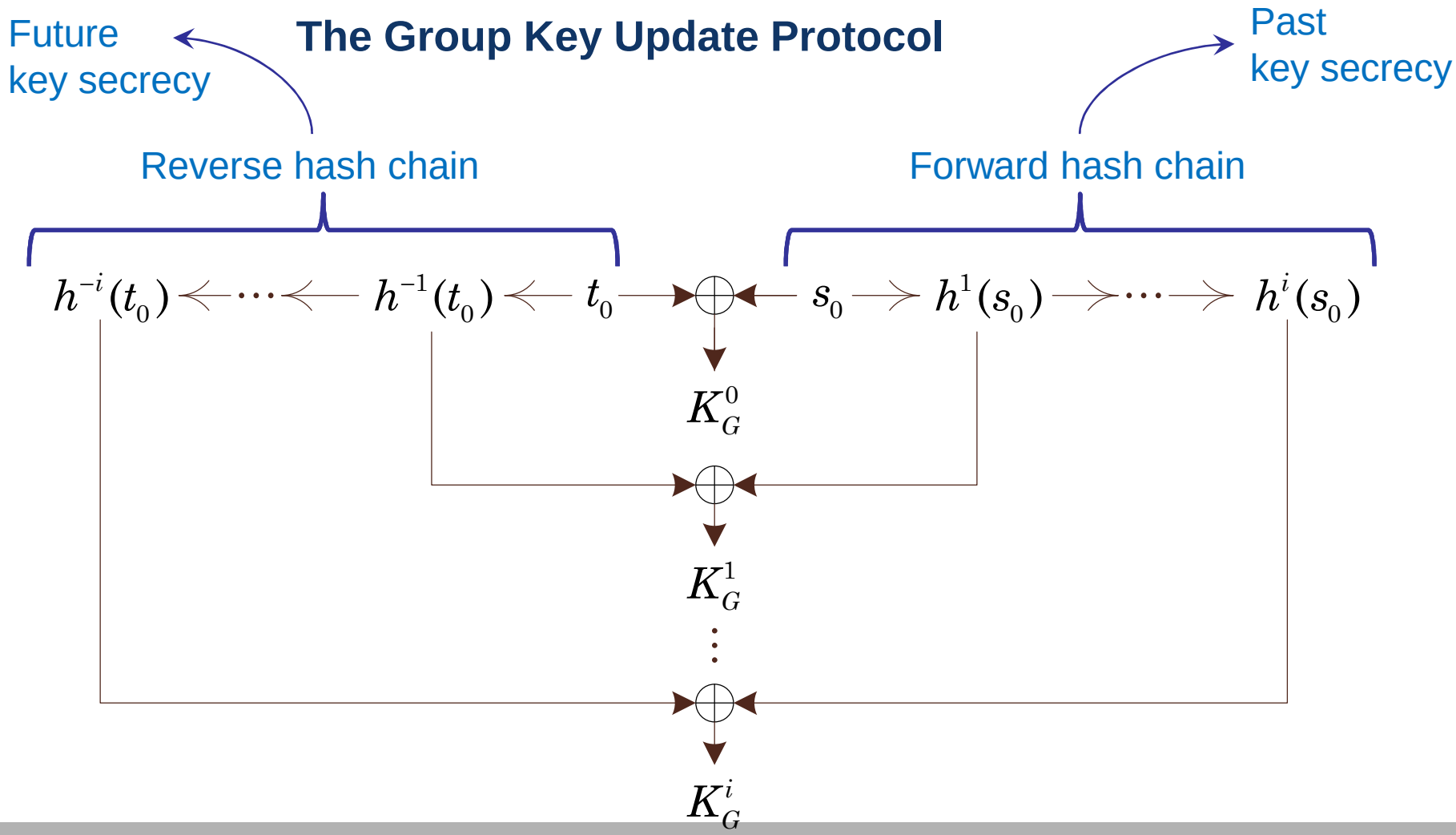
$$1.M \leftarrow N: \{r_N\}_{K_M}, MAC_{K_{MN}}(\{r_N\}_{K_M})$$

M, N: computes the new pairwise key

$$K'_{MN} = h(K_{MN}, r_N)$$

- The value of the new pairwise key K_{MN} is only determined by the sensor node.
- The key input r_N for the new pairwise key K_{MN} is not really random in their scheme

The Proposed Key Management



The Proposed Key Management

The Group Key Update Protocol

$$1.\ M \rightarrow N: \quad i, \{h^{-i}(t_0)\}_{K_{MN}}$$

$$2.\ M \leftarrow N: \ h_{K_{MN}}(K_G^i)$$

M, N: update the group key,

i.e., $K_G^i := h^i(s_0) \oplus h^{-i}(t_0)$

The Proposed Key Management

The Pairwise Key Update Protocol

1. $M \rightarrow N: \quad i, \{h^{-i}(t_0), g^{r_M}\}_{K_G^{i-1}}$ # broadcast message

2. $M \leftarrow N: \quad \{g^{r_N}\}_{K_{MN}}, h_{K_{MN}}(g^{r_M}, g^{r_N})$

M: keeps the hash value of the current pairwise key: $K_{MN} := g^{r_M r_N}$

M, N: increment the group key index from $i - 1$ to i, and update the values of the pairwise key and the group key, i.e.,

$$K_{MN} := g^{r_M r_N}, \qquad K_G^i := h^i(s_0) \oplus h^{-i}(t_0)$$

The Proposed Key Management

Delivery Failure Management

- The delivery failure in WSNs leads to key mismatch of group keys and/or pairwise keys.
- The key mismatch is a big concern here due to the absence of long term key.
- Simple transmission of protocol messages is an impractical solution:
  - It may open the door to replay attacks.
  - It may require sensor nodes to go back to the old key even after a successful update for the pairwise key.

The Proposed Key Management

Delivery Failure Management

1. $M \rightarrow N: \quad i, j, \{h^{-i}(t_0),\ g^{r_M}\}_{K^j_{M\,N}}$ # unicast message

2. $M \leftarrow N: \ \{g^{r_N}\}_{K^j_{M\,N}},\ h_{K^j_{M\,N}}(g^{r_M}, g^{r_N})$

- Key idea: Keep the *hashed copy* of pairwise key

$$K^j_{MN} = h^j(K_{MN})$$

Conclusion

- Four types of adversaries varying in their capability are derived.
- Cryptographic countermeasure alone cannot prevent the most powerful adversary in the WSN.
- Nilsson et al.’s work turned out to provide neither past key secrecy not future key secrecy against node compromise by any type of adversaries.

Conclusion

- We applied Lamport's reverse hash chain as well as usual hash chain.
- No delivery for the whole value of the new group key for group key update.
- Our scheme provides past and future key secrecies against node capture by all adversary types except type IV.
- Sandwich Attack.

Questions