

Key Management in Distributed Sensor Networking

DARPA Sensor IT Workshop
April 4, 2000

David Carman, Dr. Brian Matt,
Peter Kruus, David Balenson,
Dr. Dennis Branstad

NAI Labs, The Security Research Division
Network Associates, Inc.

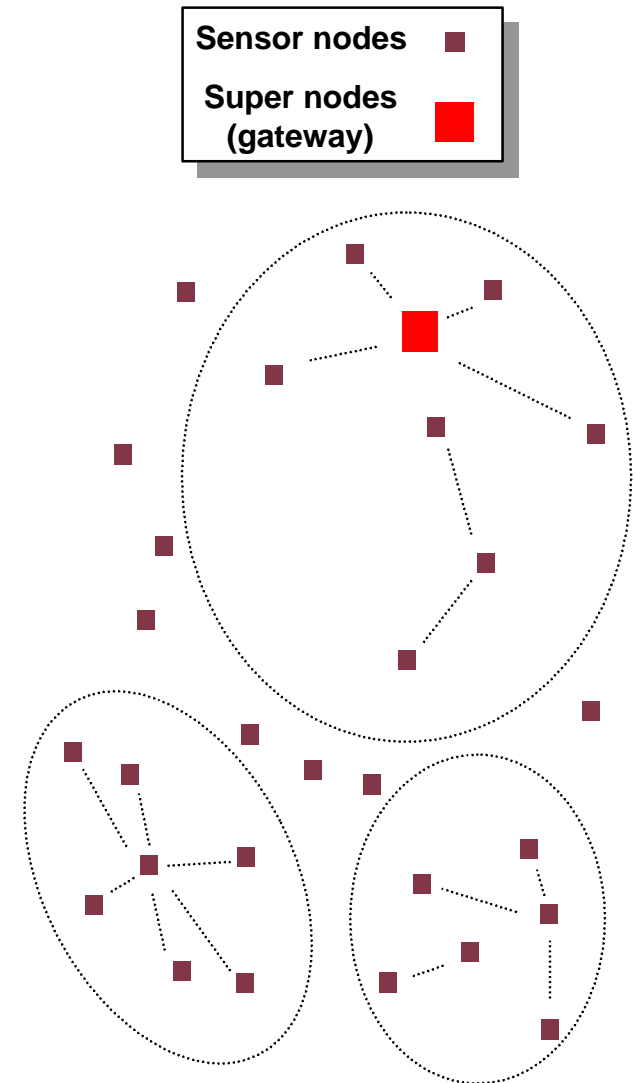
Sponsored by the
DARPA/ITO Sensor Information Technology (SensIT) Program
Through Air Force Research Laboratory (AFRL) Contract No. F30602-99-C-0185
Dr. Sri Kumar, DARPA, Program Manager
Scott Shyne, AFRL, COTR

Objective and Plan

- Objective
 - Provide energy-efficient and secure key management for confidentiality and group level authentication
 - Identify the trusted group
 - Key the trusted group
 - Protect against various threat scenarios
- Plan
 - Identify security-relevant characteristics of DSN groups
 - Identify and analyze constraints
 - Develop and analyze candidate keying approaches
 - analyze tradeoffs
 - examine use of hybrid and multiple approaches

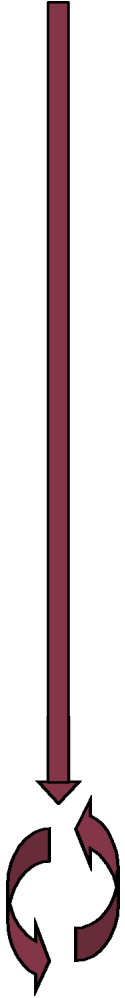
Security-Relevant Characteristics of DSN Groups

| DSN Characteristic | Security Relevance |
|--|--|
| Size: up to 1000s | Single key untenable |
| Unattended | Compromisable |
| Communications vs. computational energy costs | One keying scheme won't be optimal in all scenarios |
| Support data fusion | No end-to-end encryption |
| Unicast vs. multicast | Effects keying choices |
| Composition unknown apriori | Discourages key pre-distribution |
| Requires survivability | Group keying has points of failure |
| >99% energy-limited nodes, <1% energy-endowed "super" nodes | Limits ability to exploit energy-endowed nodes |
| Intermittent connectivity – frequent re-routing | Discourages group keying of large groups |
| Isolated sub-groups | Group keying w/o access to gateway |



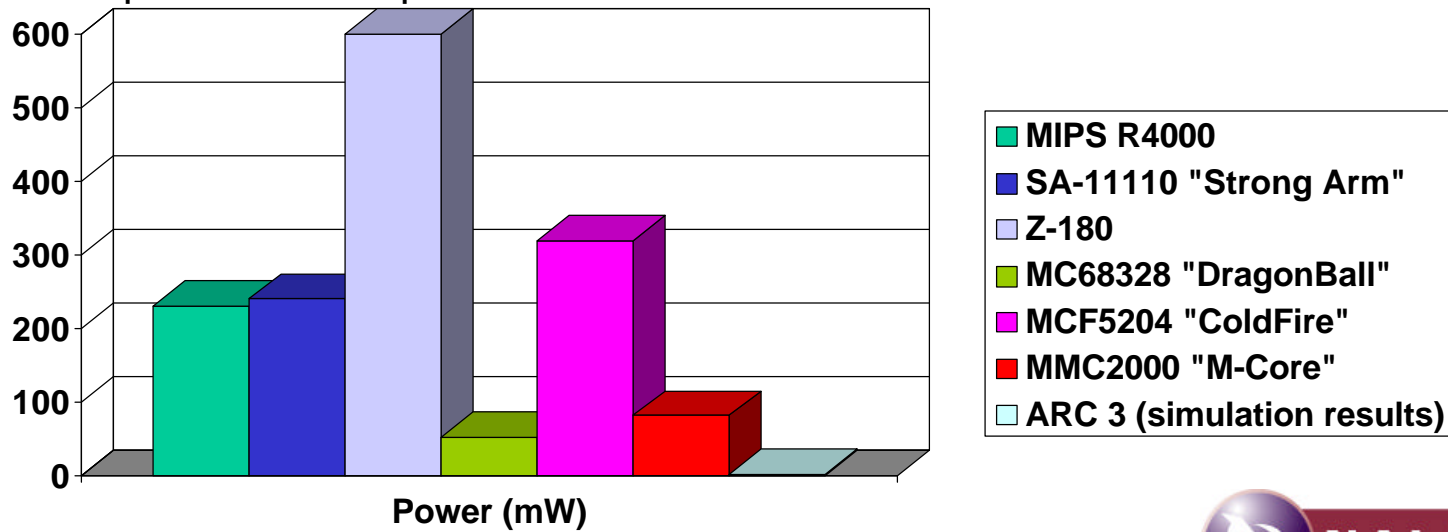
Security Concept of Operations

- Manufacturing
 - Initialize public key infrastructure
 - Hard code public keys into sensors
- Pre-deployment
 - Optionally load global and/or granular keys
 - Establish unique sensor “certificates”
- Deployment
- Routing (also called Assembly)
 - Develop long-term keying relationships with neighbors
- Sensor Applications
 - Generate and use short-term keys for data protection
- Re-routing
 - Update/add/delete keying relationships only as necessary



Energy Constraints

- Constraints: battery capacity, communication energy, computation energy
- Battery Capacity for WINS Battery Pack¹
 - 7.2 V @ 1000 mAH yields **26 kJ**
- WINS Communications Energy¹
 - Subsystem power consumption x communication time
 - Transmit: 210 mW @ 10 kbps rate = **21 μ J/bit**
 - Receive: 140 mW @ 10 kbps rate = **14 μ J/bit**
- Computation Energy
 - CPU power consumption x computation time
 - CPU power consumption:



SensIT-040400-5 ¹source Sensoria Corp.

http://www.nai.com/nai_labs/asp_set/crypto/crypt_senseit.asp

Candidate Keying Approaches

- Predeployed symmetric keying
 - Load global “mission” key -> vulnerable to global compromise
 - Load granular keys -> reduces compromise potential
- Pairwise keying
 - Each sensor performs keying with each 1-hop neighbor
 - Forwarding requires decrypt/verify/authenticate/re-encrypt
 - Keying algorithms: RSA, DH, ElGamal, ECC, XTR
- Group keying
 - Neighborhood of sensors establish single keying relationship
 - Benefit: reduces comm. and computation energy costs
 - Keying algorithms: GDH, Burmester-Desmedt, LKH, OFT
- “Rich Uncle” keying
- Attribute-based keying
- Hybrid schemes - combine two or more above

Energy Costs for Processors/Algorithms

- Energy costs for 128-bit multiply/accumulate operation

| Processor | Power Consumption (mW) | Clock Freq. (MHz) | Native Mult. Result | # clocks to compute 128-bit result | Time required (μ s) | Energy consumed (nJ) |
|----------------------|------------------------|-------------------|---------------------|------------------------------------|--------------------------|----------------------|
| MIPS R4000 | 230 | 80 | 128 | 40 | 0.50 | 115 |
| SA-11110 "StrongARM" | 240 | 133 | 64 | 60 | 0.45 | 108 |
| Z-180 | 600 | 10 | 32 | 912 | 91 | 55000 |
| MC68328 "DragonBall" | 52 | 16 | 32 | 1920 | 120 | 6200 |
| MCF5204 "ColdFire" | 320 | 33 | 32 | 304 | 9.2 | 5800 |
| MMC2001 "M-Core" | 81 | 33 | 32 | 416 | 13 | 1000 |
| ARC 3 | 2* | 40 | 32 | 168 | 4.2 | 8.4 |

- Energy costs per algorithm per processor:

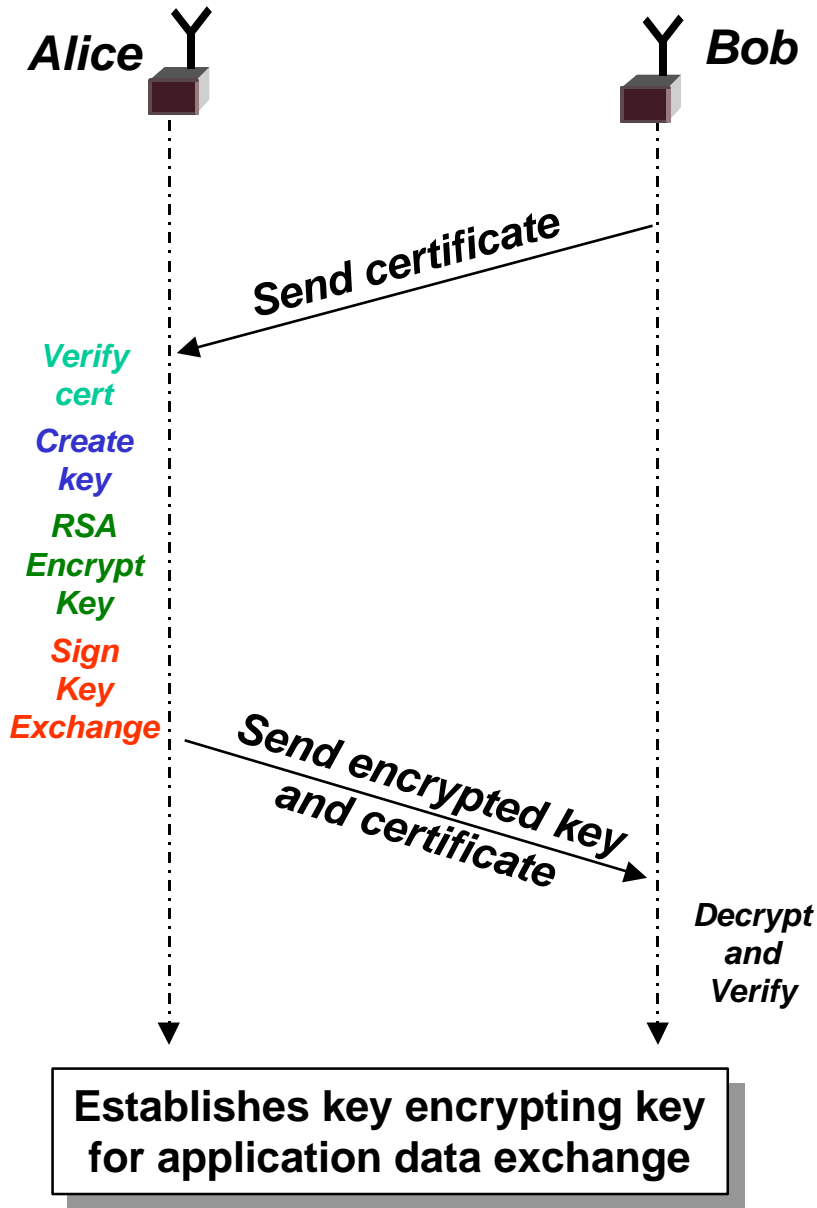
| Processor | Algorithm Computation Time and Energy (ms – mJ) | | | | | | | |
|----------------------|---|-------------|------------------|------------|--------------------|------------|------------------|-------------|
| | RSA Encrypt/Verify | | RSA Decrypt/Sign | | XTR Encrypt/Verify | | XTR Decrypt/Sign | |
| | Time(ms) | Energy(mJ) | Time(ms) | Energy(mJ) | Time(ms) | Energy(mJ) | Time(ms) | Energy(mJ) |
| MIPS R4000 | 3.5 | 0.81 | 73 | 17 | 19 | 4.5 | 8.3 | 1.91 |
| SA-11110 "StrongARM" | 3.1 | 0.74 | 62 | 15 | 17 | 4.1 | 7.1 | 1.71 |
| Z-180 | 620 | 370 | 12300 | 7400 | 3400 | 2000 | 1400 | 840 |
| MC68328 "DragonBall" | 810 | 42 | 16200 | 840 | 4465 | 232 | 1850 | 96 |
| MCF5204 "ColdFire" | 62 | 39 | 1240 | 780 | 340 | 210 | 142 | 89 |
| MMC2001 "M-Core" | 85 | 6.9 | 1700 | 138 | 470 | 38 | 194 | 15.7 |
| ARC 3 | 28 | 0.06 | 570 | 1.13 | 156 | 0.31 | 65 | 0.13 |

* simulation result

SensIT-040400-7

http://www.nai.com/nai_labs/asp_set/crypto/crypt_senseit.asp

Energy Usage Example: Pairwise Keying



- Pairwise key exchange energy cost per node:

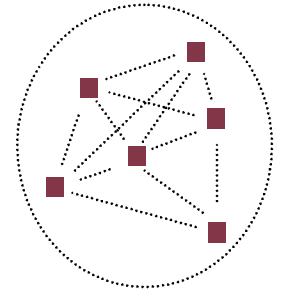
| Processor | | Energy Costs (mJ) | |
|-----------------------|-------|-------------------|-----|
| | | RSA | XTR |
| MIPS R4000 | Comm. | 114 | 57 |
| | Comp. | 18 | 9 |
| | Total | 132 | 66 |
| MC68328 "Dragon Ball" | Comm. | 114 | 57 |
| | Comp. | 903 | 443 |
| | Total | 1017 | 500 |

- Number of key exchanges if only 1% of the WINS energy is available for key management (260 J):

| Processor | Number of Key Exchanges | |
|-----------------------|-------------------------|------|
| | RSA | XTR |
| MIPS R4000 | 1970 | 3940 |
| MC68328 "Dragon Ball" | 256 | 520 |

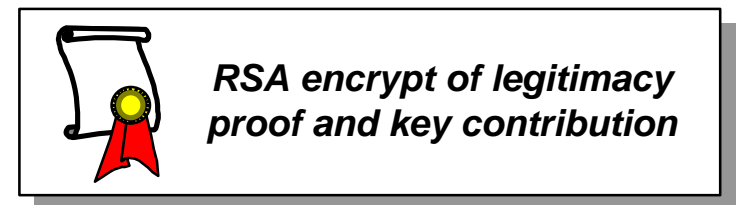
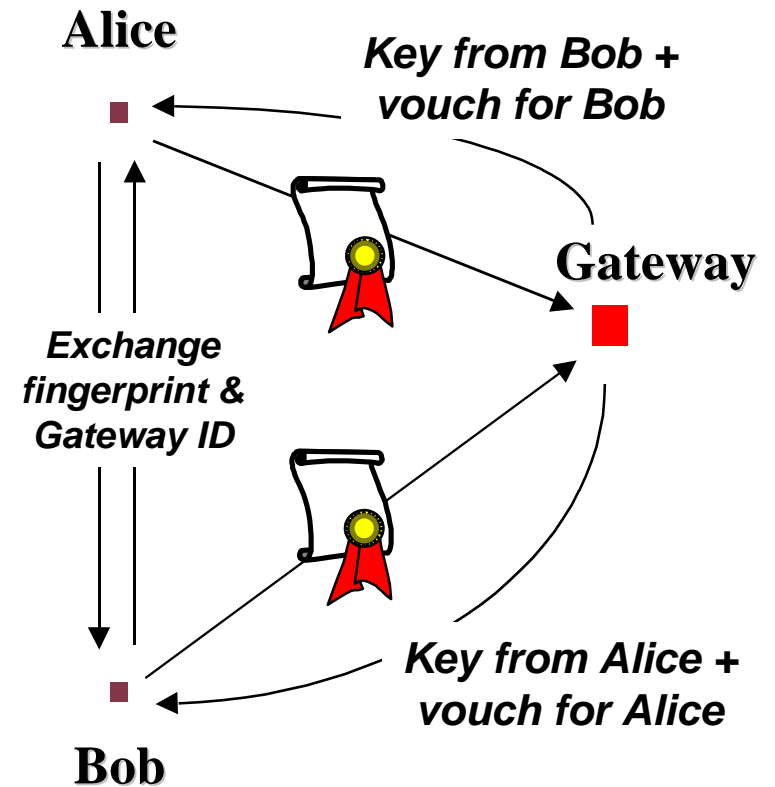
Group Keying Energy Costs

- Scenario:
 - six 1-hop connected WINS (MIPS R4000) nodes
 - transmission costs significant portion of total costs
- Pairwise:
 - Energy cost/node: $132 \text{ mJ/pair} * 5 \text{ pairs} = \underline{660 \text{ mJ}}$
- Group Keying, Unicast (GDH-IKA.2):
 - Nodes 1-4: 3 exponentiations, 2 transmissions, 2 receives
 - Node 5: 2 exponentiations, 6 transmissions, 2 receives
 - Node 6: 6 exponentiations, 5 transmissions, 6 receives
 - *Average energy cost/node: $\underline{300 \text{ mJ}}$ (55% reduction from pairwise)*
- Group Keying, Multicast (Burmaster-Desmedt):
 - All nodes perform three exponentiations, transmit two multicast msgs, and receive two multicast msgs
 - Energy cost/node: $\underline{220 \text{ mJ}}$ (27% reduction from unicast)

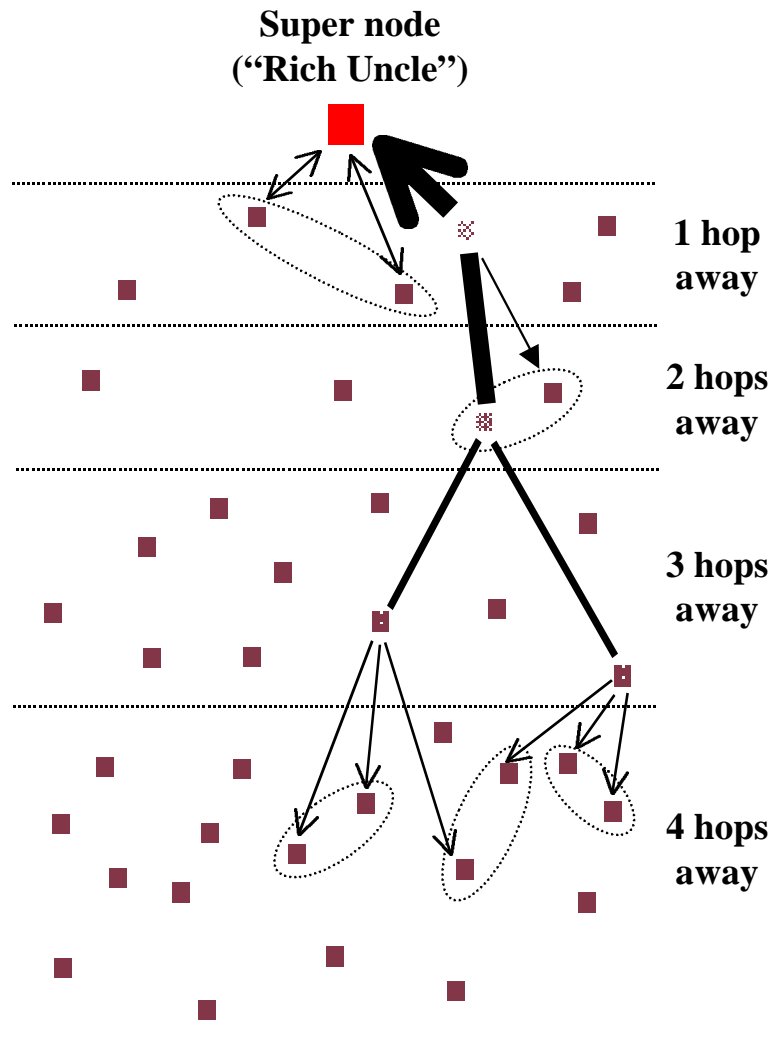


“Rich Uncle” Keying Scheme

- Energy-limited nodes offload crypto costs to energy-endowed super nodes
- Efficient when crypto energy costs > comm. energy costs (e.g. DragonBall)
- Particularly beneficial to heavily taxed nodes near an energy-endowed gateway
- Sensor node energy costs (DragonBall):
 - Pairwise RSA exchange cost per sensor node:
 - **1017 mJ**
 - Rich Uncle exchange per sensor node:
 - **453 mJ**
- “Rich Uncle” can be combined with unicast and multicast group keying for even greater benefit



Multi-hop "Rich Uncle" Keying Scheme



- Concept: extend benefits to nodes greater than one hop from gateway
- Combine group keying with multi-hop "Rich Uncle"
 - complex to determine benefits - need to simulate?

| | |
|----------------------------------|---|
| Sensor nodes | ■ |
| Energy-taxed sensor nodes | ⊠ |
| Super nodes (gateway) | ■ |

Latency

- Key management latency (prior to appl data exchange)
 - Pairwise (WINS, RSA - worst case):
 - Comm @10kbps: 0.65s , Comp: 0.16s , Total: **.81s** per keying pair
 - Group (WINS, unicast GDH - worst case):
 - Comm @10kbps : 3.0s , Comp: 6.6 , Total: **9.6s** per 6-node group
 - “Rich Uncle” (WINS, basic):
 - Comm @10kbps : 0.73s , Comp: 0.33s , Total: **1.06s** per keying pair
- Encryption/authentication latency
 - Confidentiality (using AES estimate): 5 μ s per 128-bit block
 - Authentication (using HMAC-SHA-1 estimate): 16 μ s per 512-bit block
 - Total encryption/authentication latency for 10kbit packet:
 - **0.72 ms**
 - Encryption/authentication energy cost per bit for WINS @ 10kbps:
 - **16 nJ/bit**
 - compare to 21 μ J/bit for transmission and 14 μ J/bit for reception

Summary

- **Energy** is main constraint, not power
- **Processor characteristics** and **communications costs** primarily determine key management energy costs
- **Computational energy costs** vary widely with processor
- **Group keying** offers significant reductions over pairwise when communications costs are large part of total costs
- **Multicast** capability reduces group key management energy costs in some scenarios
- “**Rich Uncle**” scheme reduces energy costs when computation costs $>$ communications costs
- **Computational latency** for both initial keying and encryption/authentication is relatively small