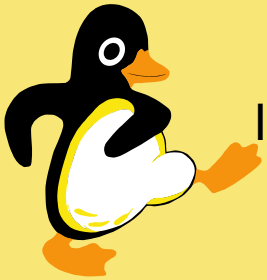


Internet of Things (IoT) Routing and Security

Dr. Ralf Schlatterbeck
Open Source Consulting

Email: office@runtux.com
Web: http://www.runtux.com
Tel. +43/650/621 40 17

Constrained Device

- 128kB (*k* not *M*) Flash (Code)
- 32kB RAM, 2k-8k EEPROM
- 16 MHz clock
- typically with IEEE 802.15.4 radio on chip
- . . . and AES encryption on chip
- Since 2003 we know Internet Protocol (IP) works
- µIP [Dun03, Wik16] (open source) NanoIP [SMR+03]
- 6LoWPAN: IPv6 over IEEE 802.15.4
- Typically UDP not TCP
- Devices are *not* getting more powerful but smaller
- Well, there *is* a newer generation of 16-bit devices

Constrained Device

"While there are constantly improvements being made, Moore's law tends to be less effective in the embedded system space than in personal computing devices: gains made available by increases in transistor count and density are more likely to be invested in reductions of cost and power requirements than into continual increases in computing power." RFC 7397 [GT14]

IoT Protocol Stack

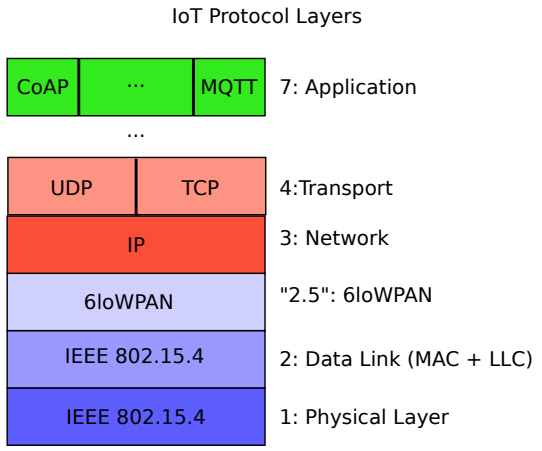


Figure 1: Example IoT Protocol Stack

BLE IoT Protocol Stack

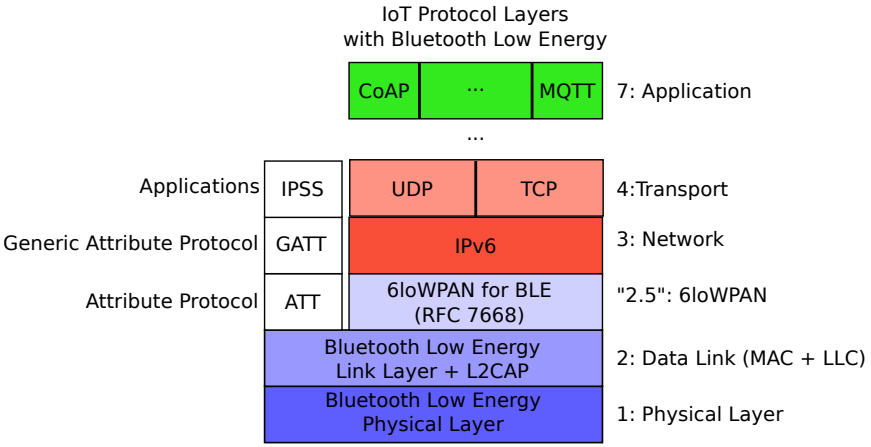
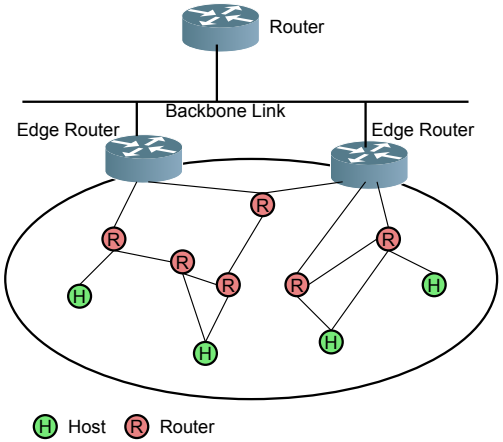


Figure 2: IoT Protocol Stack with BLE [NSI+15]

IEEE 802.15.4, 6LoWPAN



6loWPAN: Routing

- RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks RFC 6550 [BHK+12]
- IETF ROLL working group
- Routing for constrained devices and constrained links
- Claims to work for few dozen to thousands of routers
- Based on IPv6
- Uses new ICMPv6 type for its control messages
- Builds Destination oriented acyclic graphs (DODAG)
- Distributed algorithm

RPL Problems

- Standard is too loosely defined
- More “may/optional” than “must/shall” terms [HSKP17]
- Two objective functions for routing decisions
  - OF0 (objective function zero) [Thu12]
  - MRHOF (Minimum Rank with Hysteresis) [GL112]
- Use of OF is tied to border router
- Nodes would have to implement both functions
- But: e.g. Contiki implements MRHOF while RIOT implements OF0

→ Heavy interoperability problems

RPL Problems

- RPL tries to allow “sleepy” routers
- It has long been proposed (before standardization of RPL) that separating both functions is a good idea [SSSD10]
- RPL is bad in routing down (border router to node): Protocol is optimized for up-routing (sensor node)
- Route update for downward route can be very slow [HSKP17]
- . . . because RPL tries to minimize routing overhead and routing is initiated from below
- Routes can be asymmetric

RPL Problems

- Routing metric ETX (expected transmissions)
- Cannot distinguish robust link from marginal, e.g. -80dBm vs. -50dBm
- At a certain RSSI threshold ETX can drop from >90% to <10% [SDTL10]
- With WIFI interference RPL’s routing topology churns [HG13]
- Frequent route changes make the problem worse
- One empirical paper disproves many asumptions of RPL [SDTL10]
- Claim “thousands of nodes” was never fulfilled

RPL Problems

- Two routing strategies for downward routes
- . . . storing and non-storing mode
- Non-storing mode: source-routing for downward
- . . . packet travels to the root and then down again
- . . . has its own share of problems with IP encapsulation [RRT21]
- For storing mode due to the forest topology routers near the root need much routing memory
- Moderate sizes of the network run out of routing entries for small routers [HSKP17]
- Another problem is the size of the neighbor table

Google Thread

- Targets 100’s not 1000’s of nodes [KKC19]
- Routers never sleep
- Maximum of 32 routers (5 bit of address)
- Topology among routers is a full mesh not a forest
- This allows for asymmetric routes among routers
- Number of routers limits routing entries
- Steady-state route update frequency much higher than RPL (32s)
- Uses Signal-to-noise ratio (SNR) for routing, not ETX
- OpenThread is an open source implementation

Physical Security

- Relatively easy to physically obtain (and control) at least one node on the network
- Security of entire network must not depend on integrity of a single device
- Keys may be contained in the memory of a device!

→ Hard problem! [SB09]

- If only Layer-2 encryption is used, messages are in cleartext on every (intermediate) node
- Adversary who PWNs a router node sees all traffic through this node

IEEE 802.15.4 Layer 2

- Counter with CBC-MAC (CCM) mode of AES [WHF03]
- Uses a *nonce* “Number used once”
- Security depends on nonce really used only once
- 13 byte nonce:
  - 8 bytes address of device
  - 4 byte frame counter
  - 1 byte security level
- Alternative: 4-byte int. atomic time TAI, unit 2^{-10}s + 1 byte sequence number

→ Around 2^{22}s for key lifetime (< 7 weeks)

Hardware Requirements for nonce

- Non-volatile memory for frame counter
- Many devices *do* have an EEPROM
- Make sure device doesn’t come up with empty EEPROM
- Alternative: Real-Time Clock
- . . . with Battery to ensure clock moves forward
- Most devices do *not* have a battery
- Updating EEPROM with every packet sent

→ typical EEPROM guaranteed write cycles are 20’000–100’000

→ We need key management

Hardware Requirements for Key management

- Key-management protocol: Diffie-Hellman [DH76]
- There are derivatives and similar algorithms
- All need a source of randomness
- “IoT devices using TLS/DTLS must offer ways to generate quality random numbers. There are various implementation choices for integrating a hardware-based random number generator into a product: an implementation inside the microcontroller itself is one option, but dedicated crypto chips are also reasonable choices.” [Fos16, p. 39f]
- RFC 4086 gives technical background [ESC05]

Mesh Routing

- Routing Protocol for Low-Power and Lossy Networks (RPL)
- RFC 6550 [BHK+12]
- Based on IPv6
- Uses new ICMPv6 type for its control messages
- Can use link-layer security
- . . . or its own security modes
- Crypto uses AES-CCM

→ We have a nonce

→ RPL doesn't really work [KKC19]

Layer 3: IPsec

- Layer-3 is IP, IPsec for Encryption/Security
- Internet Key Exchange (IKE) Protocol [Kau05]: poor fit (too complex) for LoWPANs [SB09]
- Encapsulating Security Payload (ESP) [Ken05]
- Will probably also use CCM/AES [WHF03, SB09]
- Described in RFC 4309 [Hou05]
- ESP should not be too heavyweight [SB09]

→ IKE is too complex for constrained devices

→ We have a nonce

→ Key management needed

Layer 4: DTLS

- Datagram Transport Layer Security [RM12, ML15]
- DTLS is for UDP like TLS is for TCP
- So we have coaps like https with 's' for 'secure'
- Need easy integration not just into Contiki-OS
- OpenSSL too big/memory-hungry for embedded
- contiki-dtls Vladislav Perelman @ Jacobs University Bremen [Per12]
- tinydtls Olaf Bergmann
- mbed TLS (previously PolarSSL) (too big for 8-bit)
- tinydtls better supported than contiki-dtls
- But mbed TLS is cleaner than tinydtls

DTLS Complexity

- DTLS has too many options for embedded
- Typical Certificate Hierarchy not applicable to IoT
- . . . some algorithms too big for constrained devices

→ Asymmetric Crypto currently not in hardware for many devices!

- Stripped-down version use less options
- CoAP Standard defines mandatory and optional algorithms
- RFC 7925 specifies IoT Profiles for TLS/DTLS [Fos16]

DTLS Overhead

- Establishing DTLS needs many messages
- Many more messages for DTLS than for payload!
- DTLS-Context needs much (RAM) memory
- . . . often too much for constrained device
- Asymmetric crypto often not available
- . . . due to size of code (ROM) *and* data (RAM)
- DTLS allows pre-shared keys
- It would be beneficial if DTLS context lived longer

Encryption Layer 7 (or Layer 5?)

- Layer-7: Application Layer
- Layer-5: Presentation Layer: Better fit?!
- For IoT: CoAP
- Object Security for Constrained RESTful Environments (OSCORE) [SMPS19]
- Encrypts (part of) CoAP payload
- Some CoAP options affect networking, must stay in clear
- But only for CoAP Protocol!
- Currently our best bet for encryption for constrained devices!

Encryption Layer 7 (or Layer 5?)

- Security context is longer-lived than for DTLS
- Need not establish security context often
- Overhead is minimal
- . . . both for communication as for implementation

→ Better suited for constrained devices

- Crypto uses AES-CCM

→ We have a nonce

- Key management is in draft-status (as of 2021)
- A draft for requirements has expired [VSMG20]
- Draft for a Diffie-Hellman protocol was just issued [SMP21]

Key management

. . . "'AES CCM should not be used with statically configured keys. Extraordinary measures would be needed to prevent the reuse of a counter block value with the static key across power cycles. To be safe, implementations MUST use fresh keys with AES CCM."' [Hou05]

- Key management is required
- Not just for IPsec but all AES CCM applications
- Yet full IKE is unusable [SB09]

Key management

Two phases of key management

1. Commissioning: Initial Keys or certificates are transferred to a device
   - Probably during a manually instantiated peering mechanism [GT14, BGB12]
   - Establishes “Root of Trust” [Fos16]
2. During Session setup these keys are used to derive session keys

- Key management is considered a hard problem

Conclusion

- Cryptographic protocols on several OSI layers
  - Layer 2: 802.15.4
  - Layer 3: IPsec
  - Layer 4: DTLS: coaps
  - Layer 5/7: OSCORE
- Key management not solved on every layer
- No cross-layer key management
- Hardware requirements
  - Random number generation
  - persistent storage
  - clock

Conclusion

- Key management is a hard problem
  - Management of pre-shared information
  - . . . symmetric keys or certificates
- Decent security impossible on 8-bit devices?
  - No asymmetric crypto in hardware
  - Assymmetric crypto too large in software
  - PWNing a node is a valid scenario

Contents

Contents

Bibliography

[BGB12] Olaf Bergmann, Stefanie Gerdes, and Carsten Bormann. Simple keys for simple smart objects. In *Proceedings of the Workshop on Smart Object Security*, Paris, France, March 2012.

[BHK+12] A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, JP. Vasseur, and R. Alexander. Rpl: Ipv6 routing protocol for low-power and lossy networks. RFC 6550,

Bibliography

Internet Engineering Task Force, March 2012.

[DH76] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, November 1976.

[Dun03] Adam Dunkels. Full TCP/IP for 8 bit architectures. In *Proceedings of the First ACM/Usenix International Conference on Mobile Systems, Applications and Ser-*

Bibliography

vices (MobiSys 2003), San Francisco, May 2003.

[ESC05] D. Eastlake, J. Schiller, and S. Crocker. Randomness requirements for security. RFC 4086, Internet Engineering Task Force, June 2005.

[Fos16] T. Fossati. Transport layer security (TLS) / datagram transport layer security (DTLS) profiles for the internet of things. RFC

Bibliography

7925, Internet Engineering Task Force, July 2016.

[GL112] The minimum rank with hysteresis objective function. RFC 6719, Internet Engineering Task Force, September 2012.

[GT14] J. Gilger and H. Tschofenig. Report from the smart object security workshop. RFC 7397, Internet Engineering Task Force, December 2014.

Bibliography

[HG13] Dong Han and Omprakash Gnawali. Performance of RPL under wireless interference. *IEEE Communications Magazine*, 51(12):137–143, December 2013.

[Hou05] R. Housley. Using advanced encryption standard (AES) CCM mode with IPsec encapsulating security payload (ESP). RFC 4309, Internet Engineering Task Force, December 2005.

[HSKP17] David E. Culler Hyung-Sin Kim, Jeong-

Bibliography

Gil Ko and Jeongyeup Paek. Challenging the IPv6 routing protocol for low-power and lossy networks (RPL): A survey. *IEEE Communications Surveys & Tutorials*, 19(4):2502–2525, 2017.

[Kau05] Internet key exchange (IKEv2) protocol. RFC 4306, Internet Engineering Task Force, December 2005.

[Ken05] S. Kent. IP encapsulating security payload (ESP). RFC 4303, Internet Engineering

Bibliography

Task Force, December 2005.

[KKC19] Hyung-Sin Kim, Sam Kumar, and David E. Culler. Thread/OpenThread: A compromise in low-power wireless multihop network architecture for the internet of things. *IEEE Communications Magazine*, 57(7):55–61, July 2019.

[ML15] B. Moeller and A. Langley. TLS fallback signaling cipher suite value (SCSV) for preventing protocol downgrade attacks. RFC

Bibliography

7507, Internet Engineering Task Force, April 2015.

[NSI+15] J. Nieminen, T. Savolainen, M. Isomaki, B. Patil, Z. Shelby, and C. Gomez. IPv6 over bluetooth(R) low energy. RFC 7668, Internet Engineering Task Force, October 2015.

[Per12] Vladislav Perelman. Security in IPv6-enabled wireless sensor networks: An implementation of TLS/DTLS for the contiki

Bibliography

operating system. Master thesis, Jacobs University Bremen, June 2012.

[RM12] E. Rescorla and N. Modadugu. Datagram transport layer security version 1.2. RFC 6347, Internet Engineering Task Force, January 2012.

[RRT21] Using RPI option type, routing header for source routes, and IPv6-in-IPv6 encapsulation in the RPL data plane. RFC

Bibliography

9008, Internet Engineering Task Force, April 2021.

[SB09] Zach Shelby and Carsten Bormann. *6LoWPAN*. Wiley Series in Communications Networking & Distributed Systems. Wiley, Chichester, UK, 2009.

[SDTL10] Kannan Srinivasan, Prabal Dutta, Arsalan Tavakoli, and Philip Levis. An empirical study of low-power wireless. *ACM Trans-*

Bibliography

actions on Sensor Networks, 2(6):16:1–49, March 2010.

[SMP21] G. Selander, J. Mattsson, and F. Palombini. Ephemeral diffie-hellman over COSE (EDHOC). Internet-draft, expires 2021-10, Internet Engineering Task Force, April 2021.

[SMPS19] G. Selander, J. Mattsson, F. Palombini, and L. Seitz. Object security for constrained RESTful environments (OS-

Bibliography

CORE). RFC 8613, Internet Engineering Task Force, October 2019.

[SMR+03] Zach Shelby, P. Mähönen, J. Riihijärvi, O. Raivio, and Pertti Huuskonen. NanoIP: The zen of embedded networking. In *Proceedings of the IEEE International Conference on Communications (ICC 03)*, Anchorage, Alaska, May 2003.

[SSSD10] Thomas Schmid, Roy Shea, Mani B. Srivastava, and Prabal Dutta. Disentangling

Bibliography

wireless sensing from mesh networking. In *Proceedings of the 6th Workshop on Hot Topics in Embedded Networked Sensors*, pages 3:1–5, June 2010.

[Thu12] Objective function zero for the routing protocol for low-power and lossy networks (RPL). RFC 6552, Internet Engineering Task Force, March 2012.

[VSMG20] M. Vucinic, G. Selander, J. Mattsson, and D. Garcia. Requirements for a lightweight

Bibliography

AKE for OSCORE. Internet-draft, expired 2020-12, Internet Engineering Task Force, June 2020.

[WHF03] D. Whiting, R. Housley, and N. Ferguson. Counter with CBC-MAC (CCM). RFC 3610, Internet Engineering Task Force, September 2003.

[Wik16] μIP (micro IP). Wikipedia article, Wikipedia, 2016. Accessed 2016-10-13.