Cryptography for Highly Constrained Networks

—

René Struik

(Struik Security Consultancy)

e-mail: rstruik.ext@gmail.com

Outline

1. Highly Constrained Networks
   - Examples & Use Case Scenarios
   - Constraints
2. Key Management
   - Certificate Lifecycle
   - Certificate Revocation Lists
   - Authentication vs. Authorization
   - Managing Symmetric Keys
3. Efficient Crypto Constructs
   - Elliptic Curves
   - Crypto Modes of Operation
4. Putting Trust in Untrusted Devices
   - Conventional Measures
   - Exploiting “Network Effects”
5. Conclusions & Future Directions

Highly Constrained Networks

- Examples & Use Case Scenarios
- Constraints

Wheeling-Pittsburg Steel Corporation

Photo courtesy Dust Networks

The Economist

APRIL 28TH–MAY 4TH 2007 www.economist.com

Will Africa ever get it right?

In praise of Yeltsin

The world's biggest banking battle

Australia's water crisis

When everything connects

A 14-page special report on the coming wireless revolution

"We're ripe, pick us!"
"I'm here, Mummy: N 51 30. 24 W 008. 19"
"Time for walkies"
"You left me here"
"I'm all out of milk"
"Blood pressure too high"
"Accident ahead"
"We're 50% off"
"I'm sensing contamination"
"Send me energy"
"Ground needs watering"

The Promise of Wireless

The Economist, April 28, 2007

Examples of Sensor and Control Networks

- Consumer Electronics
- PC Peripherals, Toys, and Gaming
- Industrial Process Control & Factory Automation
- Smart Metering
- Building Automation & Control (HVAC)
- Supply Chain Management
- Asset Tracking & Localization
- Homeland Security
- Environmental Monitoring
- Healthcare & Remote Patient Monitoring

Catch phrase: “*Internet of Things*”
2008: more “things” connected to Internet than people
2020: est. more than 31B [1] -50B [2] interconnected objects

Benefit wireless industrial sensors [3]:
♦ Efficiency gain: 25% ♦ emission reduction: 10% ♦ significant reduction ‘wiring cost’

[1] Intel (September 11, 2011);
[2] Cisco (July 15, 2011);
[3] US DOE Roadmap (2006)

Wireless Networking Standards

Wireless Local Area Networks (WLANs)

- IEEE 802.11 family (WiFi Alliance)
- Mesh Networking (802.11s)
- Fast Authentication (802.11ai)
- WiFi Alliance

Wireless Personal Area Networks (WPANs)

- 802.15.1 (Bluetooth Alliance)
- 802.15.4 (ZigBee Alliance, Wireless HART, ISA SP100.11a)
- 802.15.6 (“Body Area Networks”)
- Bluetooth ‘Lite’
- Body Area Networks

Networking IETF:

- Routing (RoLL), Applications (CoRE), Home Area Networking (HomeNet)

Other:

- Ubiquitous Computing
- DRM, Networked Gaming
- NFC Forum
- e-Payments

[…]

Constraints (1)

Constraints for Sensor Networks

High throughput is not essential, but rather

- Low energy consumption:

  Lifetime of 1 year with 2 AAA batteries (@750 mAh, 2V) yields 85μA average power consumption, thus forcing ‘sleepy’ devices (802.15.4 uses 40-60 mW for Tx/Rx)
- Low manufacturing cost:

  Low cost devices force small memory, limited computing capabilities (clock frequency: 4-16 Mhz; 10-32 kbytes ROM, 1-4 kbytes RAM, possibly no flash)

Constraints for Adhoc Networks

- No centralized management:

  No online availability of fixed infrastructure (so, decentralized key management)
- Promiscuous behavior:

  Short-lived communications between devices that may never have met before (so, trust establishment and maintenance difficult)
- Unreliability:

  Devices are cheap consumer-style devices, without physical protection (so, no trusted platform on device)

Constraints (2)

Security Constraints for Adhoc Networks

- <u>Decentralized key management:</u>
Due to no online availability fixed infrastructure, but also very ‘sleepy’ nodes
- <u>Flexible configuration and trust management:</u>
Due to promiscuous, adhoc behavior, but also survivability requirements
- <u>Low impact of key compromise:</u>
Due to unavailability of trusted platform (tamper-proofing, etc.)
- <u>Automatic lifecycle management:</u>
Due to virtual absence of human factor, after initialization

Security Design Constraints for Sensor Networks

- Implementation efficiency: protocols should use similar cryptographic building blocks
- Parallelism: design protocols have the similar message flows
- Low communication overhead: protocols must avoid message expansion if possible

Key Management

- Certificate Lifecycle
- Certificate Revocation Lists
- Authentication vs. Authorization
- Managing Symmetric Keys

Certificates

Certificate structure: $\mathit{Cert}_{\mathrm{CA}}(\mathit{Id}_{\mathrm{A}}, Q_{\mathrm{A}}, \mathit{KeyInf}\mathrm{o}_{\mathrm{A}})$

- Authentication: Binding of entity Id_{A} to public key Q_{A}, vouched for by CA
- Authorization: Binding of public key and entity to other info ($\mathit{KeyInf}\mathrm{o}_{\mathrm{A}}$), such as
  - *Key validity period*: time interval outside which key is invalid;
  - *Key usage information*: scheme key may be used with (ECDH, ECDSA, etc.);
  - *Key policy information*: certificate chain info, trust anchors, etc.
  - *User application binding*: binding key to application use, role in company, etc.

Certificate revocation reasons:

- Key compromise (change of authentication)
- Organizational change, policy change (change of authorization)

Change in *authorization* much more prevalent than change in *authentication*

Note: key revocation applies also to symmetric keys, but usually ignored there (sic!)

Certificate Revocation Lists (CRLs), Online Certificate Status Protocol (OCSP)
– Less suited, due to lack of connectivity with sensors (perhaps, even no use at all...)

Short-Lived Certificates

Computational cost may be insignificant:

<u>Example:</u>

- 30 billion interconnected objects (est. total for 2020)
- Certificates involve ECDSA, with per-signature cost of 1 scalar multiplication
- Computational cost (in HW): 100μJ; time latency: 1 second

Total energy cost: 30 billion × 100μJ
$= 3 \times 10^{10} \times 10^{-4}$ J
= 3,000 kJ ≈ 1 kWh ($0.10)

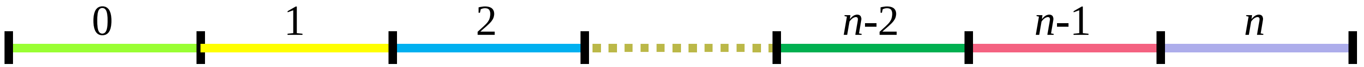
Total time: 30 billion × 1 second ≈ 1,000 years
– with 1 million RFID chips in parallel: ≈ 1/3 day = 8 hrs

Total cost: 1 million RFID chips @ $1/chip = $1 million dollars
– *per*-certificate cost: $0.00003 (assuming no re-use!)

Short-Lived Certificates at Reduced Cost

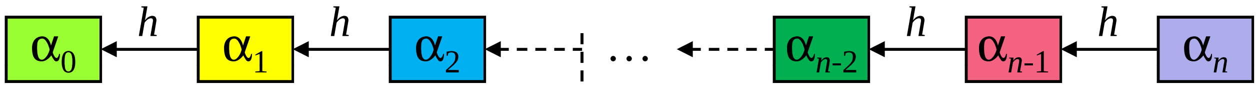
<u>Main idea:</u> (due to Micali)

- Partition time validity period in *n* intervals



- Modify Lamport's One-Way Password Scheme, so as to apply to certificates:

(a) Define hash chain:



(b) Define certificates:

– CA produces $Cert_{\mathrm{CA}}(Id_{\mathrm{A}}, Q_{\mathrm{A}}, \alpha_0)$;

– Certificate for time interval *i*: $Cert_{\mathrm{CA}}(Id_{\mathrm{A}}, Q_{\mathrm{A}}, \alpha_i)$;

– CA hands out α_1, α_2, … to A *only if* certificate still valid for interval 1, 2, …

– Verification of $Cert_{\mathrm{CA}}(Id_{\mathrm{A}}, Q_{\mathrm{A}}, \alpha_i)$ at time interval *i*:

(i) Compute $\alpha_0 = h^i(\alpha_i)$;

(ii) Substitute α_i in $Cert_{\mathrm{CA}}(Id_{\mathrm{A}}, Q_{\mathrm{A}}, \alpha_i)$, to obtain $Cert_{\mathrm{CA}}(Id_{\mathrm{A}}, Q_{\mathrm{A}}, \alpha_0)$;

(iii) Verify $Cert_{\mathrm{CA}}(Id_{\mathrm{A}}, Q_{\mathrm{A}}, \alpha_0)$

Computational cost: 1× (*vs. n*×) certificate generation, 1 hash chain computation

Efficient Crypto Constructs

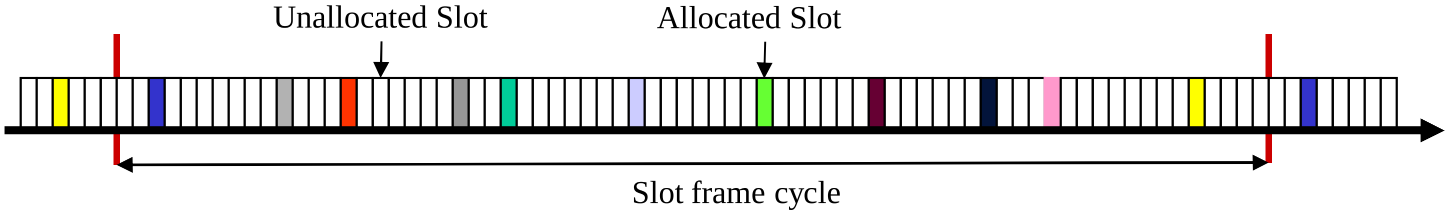
– Elliptic Curves
– Crypto Modes of Operation

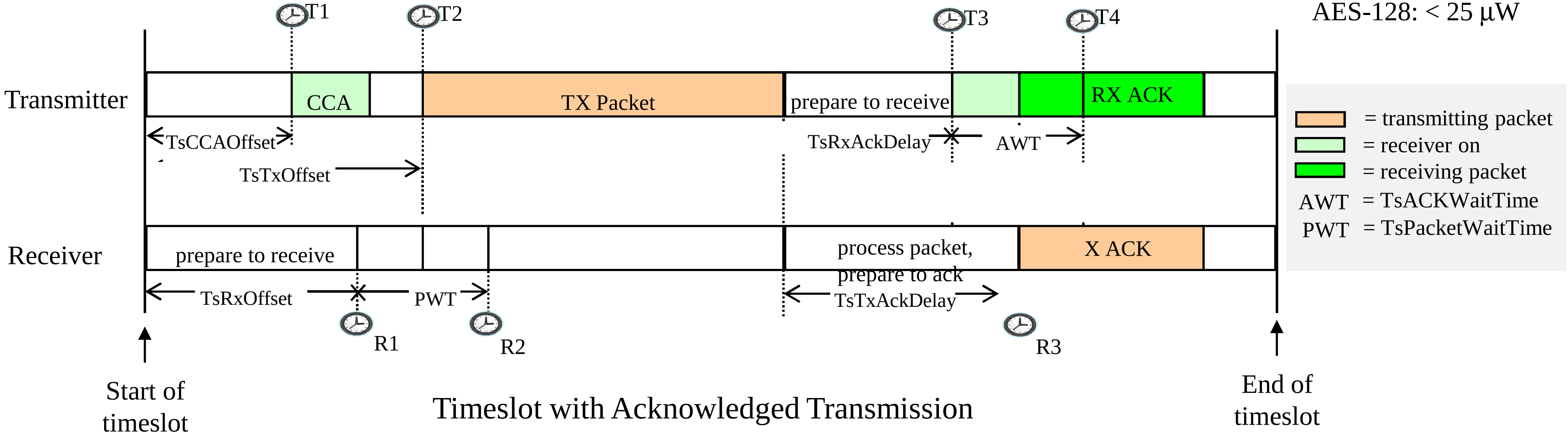
Communication and Computational Overhead Matters

Example: IEC 62951 (w/HART)

Data rate: 250 kbps
Max time jitter: 1 ms
–best in class: 0.2 ms
Power: 10 mW
Energy: 0.32µJ/octet
Latency: 32µs/octet

AES-128: < 25 µW





Timeslot with Acknowledged Transmission

Typical frame: 60 octets. Cost: 2,120µs = 200µs (listen) + 1,920µs (60×32µs) = 21.2 µJ
Communication cost savings: 8 octets = 256µs latency=2.56µJ (+14% energy efficiency)
Computational cost (in HW): AES-128 ≈ 0.2µJ; B-163 scalar multiply ≈ 20µJ-250µJ

Trade-off: Reduced communication cost ↔ Increased computational cost (& latency)

Elliptic Curves

Are we using the right curves?

- FIPS 140-2 evaluation suggests almost everyone focusing on *prime curves*
- Technical literature suggests that *binary curves* are better fit

<u>Implementation cost:</u>

Lack of data on prime curves; binary curves with very low implementation footprint

- B-163 scalar multiply ≈ 20μJ-250μJ (in HW)

<u>Computational complexity:</u>

New instruction sets (e.g., Intel’s) make binary field arithmetic very efficient

<u>Side channel resistance:</u>

Binary curves seem less susceptible to side channels (or easier to thwart):

- Goubin’s attack does apply to prime curves (e.g., P-256), but not to Koblitz curves
- Sign change attack mostly applies to prime curves
- Fault attacks yielding points of low order less applicable to binary curves

<u>Hashing into curve:</u>

Binary curves allow efficient deterministic hashing, prime curves *not* necessarily

Note: Radio engineers familiar with polynomial circuitry (such as CRC-16)

Light-Weight Crypto Mode of Operation

Are we focusing on the right problem?

Light-weight crypto:

- Focus on low-footprint, low-latency ciphers (Present, Hummingbird, etc.)
- From energy consumption perspective, mode of operation more important

Typical frame: 60 octets. Cost: 2,120μs = 200μs (listen) + 1,920μs (60×32μs) = 21.2 μJ
Communication cost savings: 8 octets = 256μs latency=2.56μJ (+14% energy efficiency)
Computational cost (in HW): AES-128 ≈ 0.2μJ; B-163 scalar multiply ≈ 20μJ-250μJ

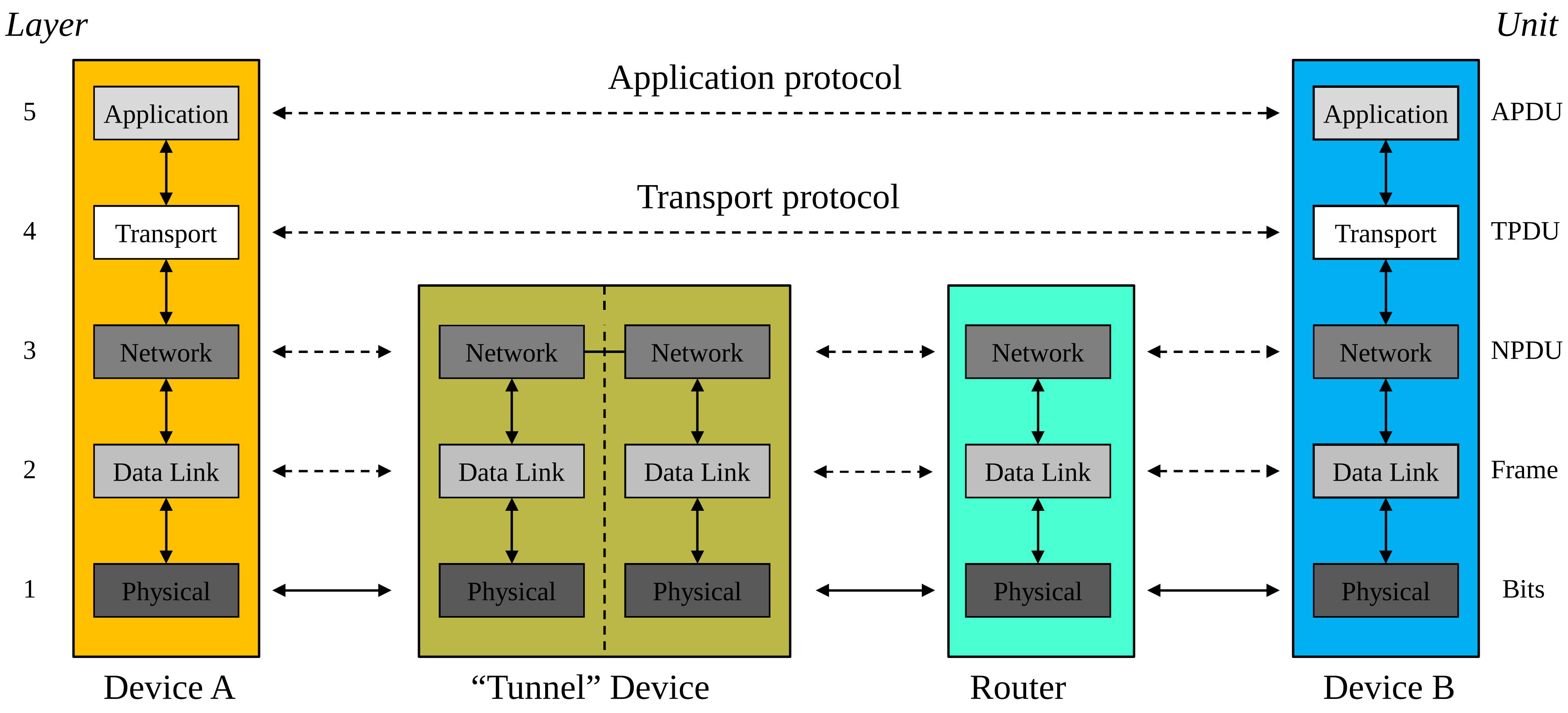
Cost of crypto: 1% of communication cost

Trade-off: Reduced communication cost ↔ Increased computational cost (& latency)

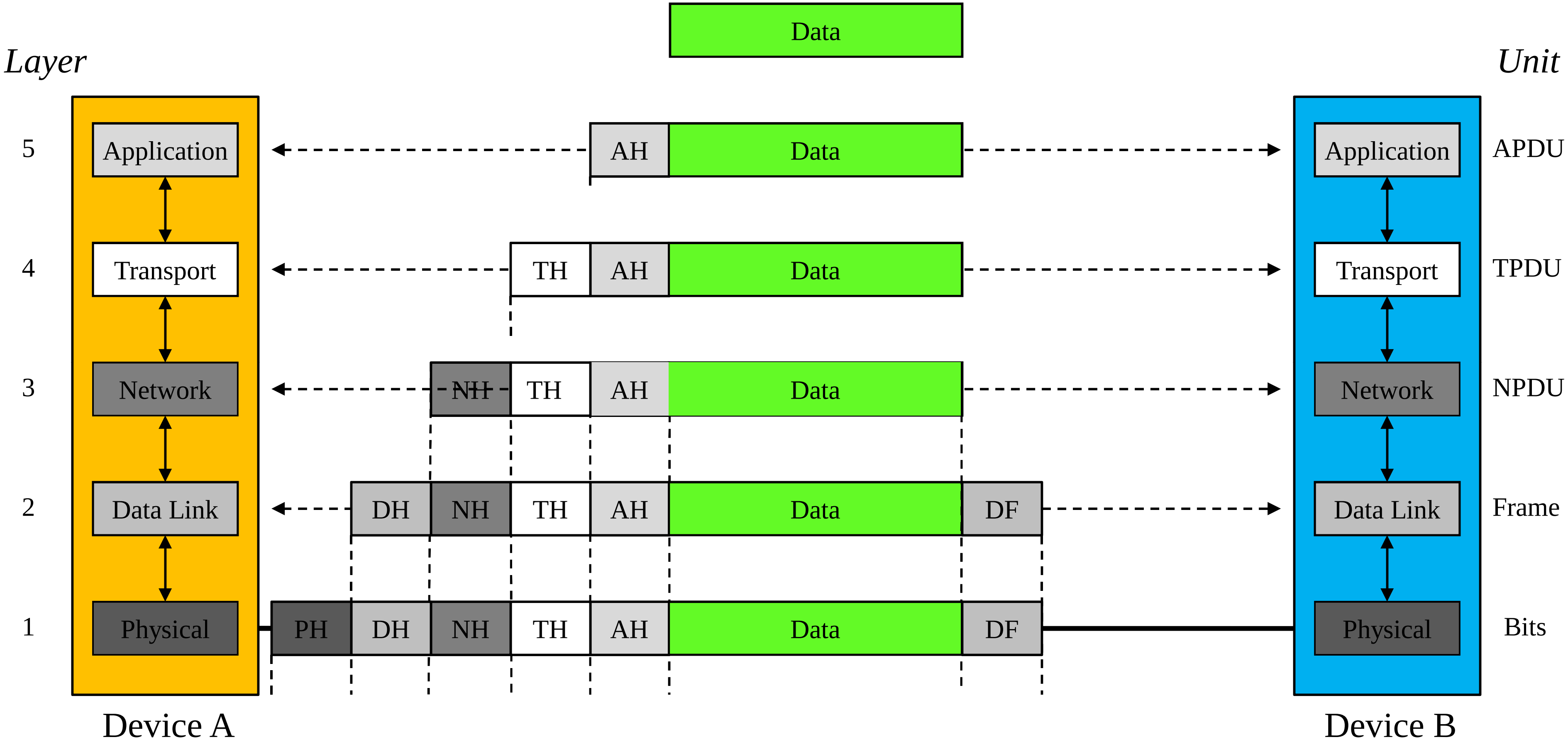
Example:

- Shaving off 8 octets may justify making symmetric-key crypto 10× more expensive

Network Layering, Protocols, Interfaces



Network Layering, without Crypto



Layer

Unit

Data

5 Application ← AH Data → Application APDU

4 Transport ← TH AH Data → Transport TPDU

3 Network ← NH TH AH Data → Network NPDU

2 Data Link ← DH NH TH AH Data DF → Data Link Frame

1 Physical PH DH NH TH AH Data DF Physical Bits

Device A

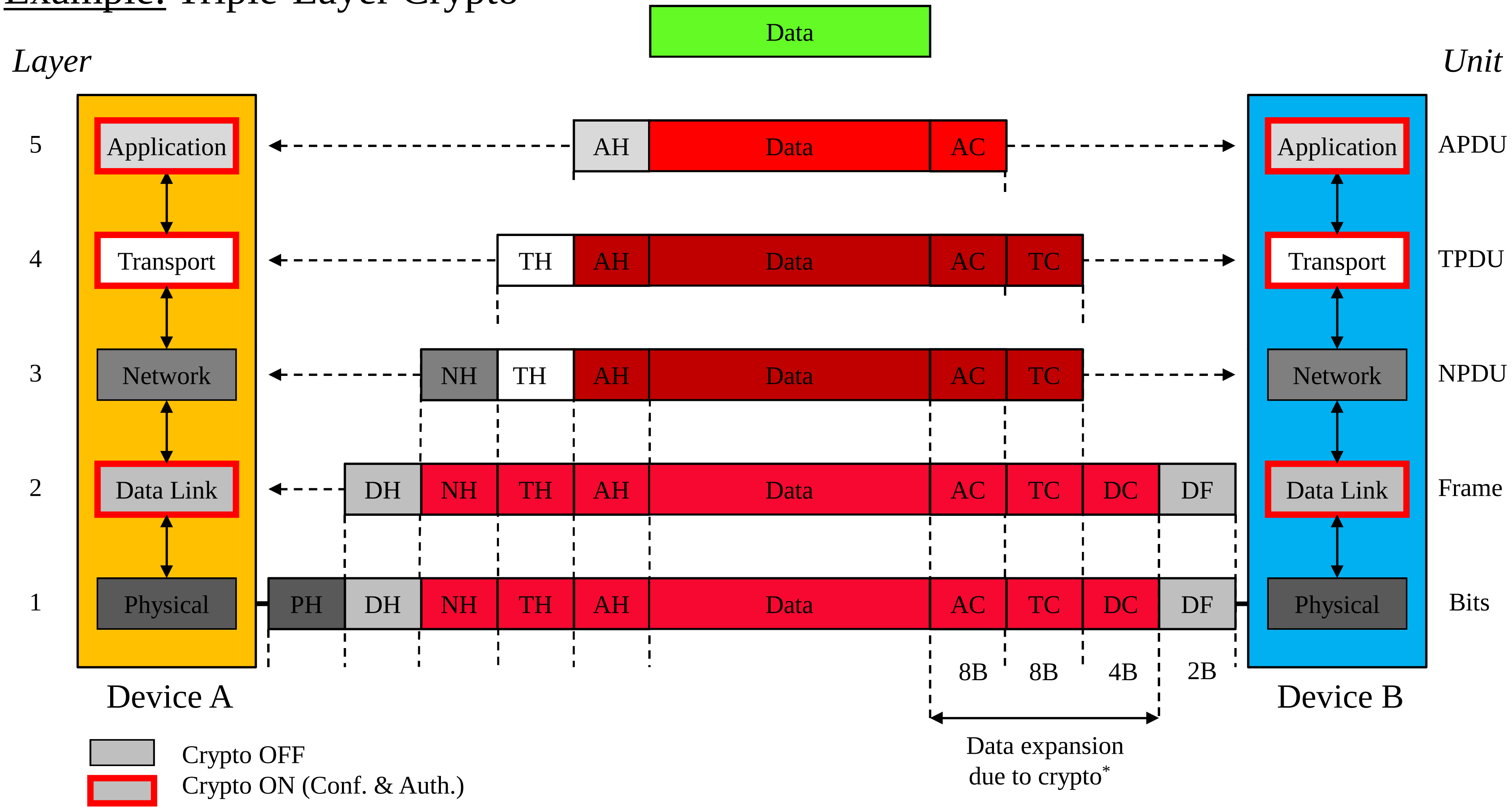
Device B

▭ Crypto OFF

🟥 Crypto ON (Conf. & Auth.)

Network Layering, with Traditional Crypto

Example: Triple-Layer Crypto



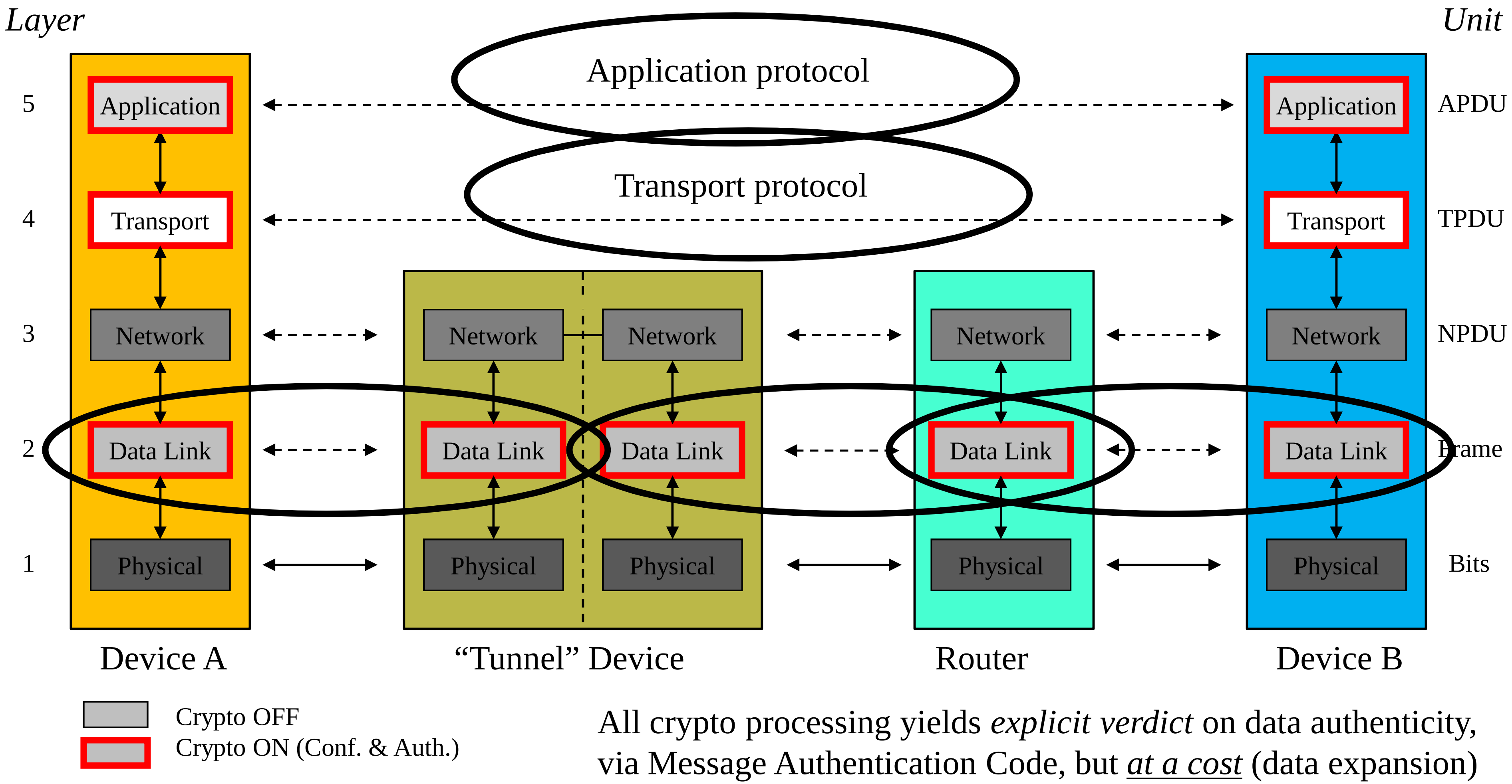
▭ Crypto OFF

🟥 Crypto ON (Conf. & Auth.)

*ignoring security admin in headers

Network Communications, with Traditional Crypto

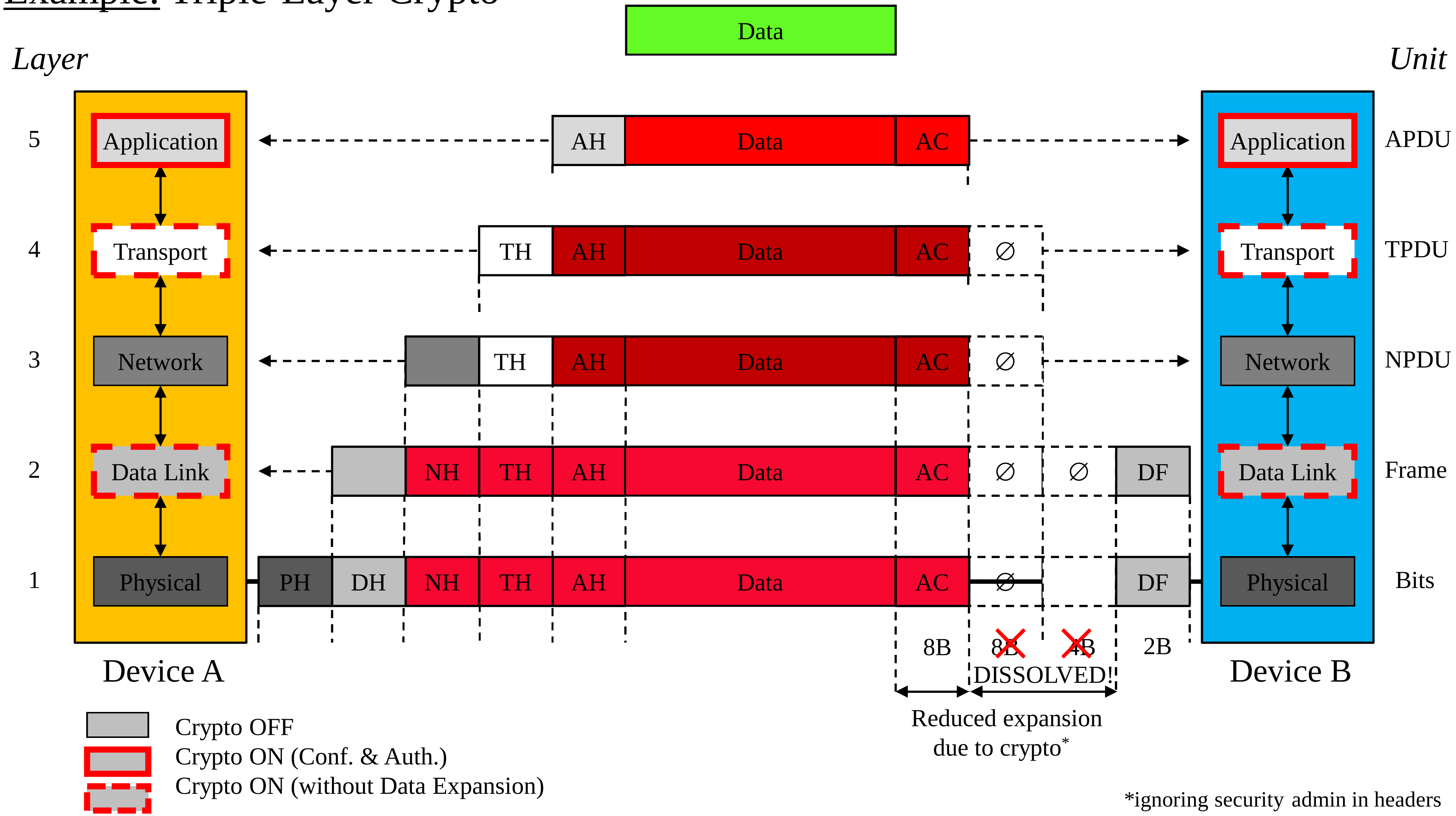
Example: Triple-Layer Crypto



All crypto processing yields *explicit verdict* on data authenticity, via Message Authentication Code, but *at a cost* (data expansion)

Network Layering, with “NEW” Crypto

Example: Triple-Layer Crypto



▭ Crypto OFF

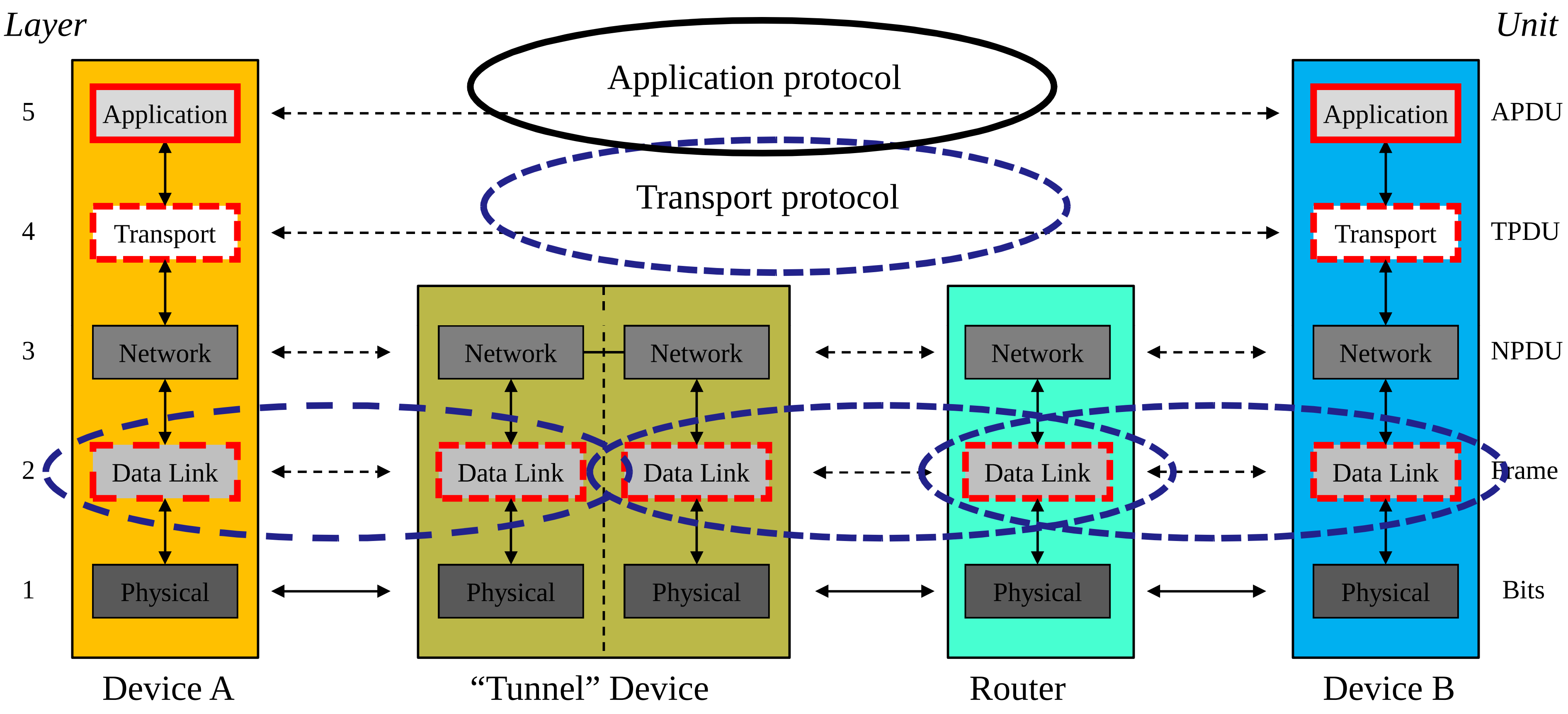
🟥 Crypto ON (Conf. & Auth.)

▨ Crypto ON (without Data Expansion)

*ignoring security admin in headers

Network Communications, with “NEW” Crypto

Example: Triple-Layer Crypto



Layer

Unit

5 Application APDU

4 Transport TPDU

3 Network NPDU

2 Data Link Frame

1 Physical Bits

Application protocol

Transport protocol

Device A

“Tunnel” Device

Router

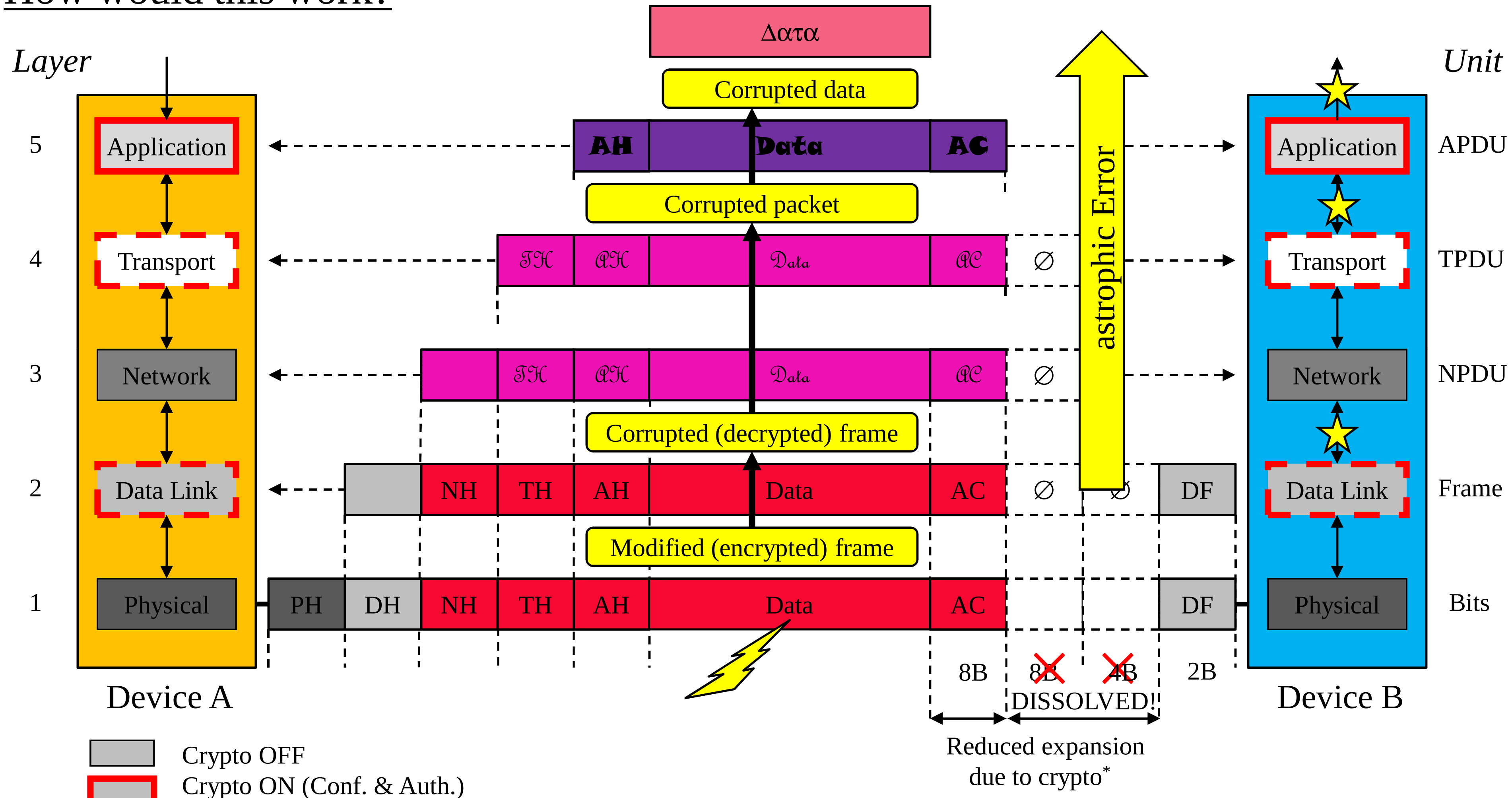
Device B

- ▭ Crypto OFF
- 🟥 Crypto ON (Conf. & New Auth.)
- ▨ Crypto ON (without Data Expansion)

All crypto processing yields *implicit verdict* on data authenticity, via *catastrophic error* produced as soon as data is modified

Incoming Processing, with “NEW” Crypto

How would this work?



Layer

5 Application

4 Transport

3 Network

2 Data Link

1 Physical

Device A

Δατα

Corrupted data

AH Data AC

Corrupted packet

TH AH Data AC ∅

TH AH Data AC ∅

Corrupted (decrypted) frame

NH TH AH Data AC ∅ ∅ DF

Modified (encrypted) frame

PH DH NH TH AH Data AC DF

8B ~~8B~~ ~~4B~~ 2B

DISSOLVED!

Reduced expansion due to crypto*

astrophic Error

Unit

Application APDU

Transport TPDU

Network NPDU

Data Link Frame

Physical Bits

Device B

▭ Crypto OFF

🟥 Crypto ON (Conf. & Auth.)

▨ Crypto ON (without Data Expansion)

*ignoring security admin in headers

"New" Crypto Mode of Operation

Applications to cryptographic protocol layering

- Significant reduction in cryptographic data expansion at lower layers
- No[1] cryptographic rejection of modified packets "in flight"
- Still possible to reject corrupted packets "in flight", if protocol layers have built-in redundancy that can easily be checked (usually the case, due to header info, etc.)

Example: ZigBee *per-packet* Security Overhead Reduction

Total security expansion ZigBee: 34 octets = 22 (NWK layer) + 12 (APL layer)

– Reduction of per-packet crypto/security overhead, to *at most* 8 octets in total only

– Potential for significant other header overhead reduction (non-security-related)

Much more payload data left for application data (≈50% more, without fragmentation)

Caveat: Cannot be realized with existing CCM* mode of operation implementation

Other applications: "storage encryption", "key wrap"

Cryptographic property: Encryption with Authenticity from Redundancy in Plaintext

Current work: (a) Plaintext larger than block-size; (b) inverse block-cipher required

[1] Some cryptographic rejection possible, if some redundancy sprinkled-in (e.g., by padding with fixed 16-bit string)

Putting Trust in Untrusted Devices

– Conventional Measures

– Exploiting “Network Effects”

Putting Trust in Untrusted Devices (1)

Conventional Measures

- Trusted implementation of crypto, including side channel resistance

What to do if devices are *intrinsically* untrusted?

- Devices are cheap consumer-style devices, without physical protection (so, no trusted platform on device)
- Still does not protect against capturing device and extracting info

Potential approach:
Use physical unclonable functions (PUFs)

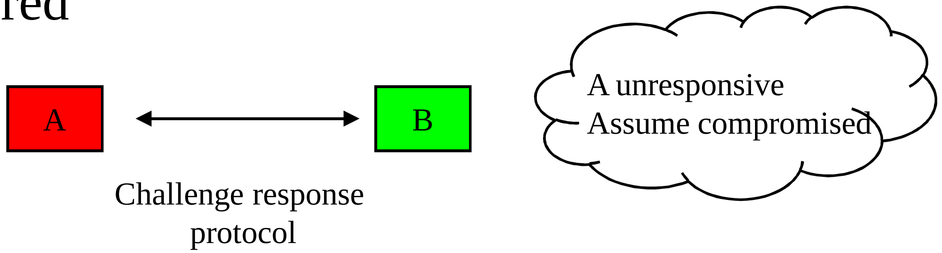
Other approach: exploit "Network effects":

- Main observation: physical extraction of keying material costs time
- Detection: captured devices may be *non*-operational during attack time window.
- Approach: Let devices "ping" each other at higher frequency than time it takes to extract keying material from captured devices.

This allows detection of potential key compromise, and corrective action (key updates)

Putting Trust in Untrusted Devices (2)

<u>Example:</u> 2 nodes, one captured



Generalizations:

Define random pairs (or subsets) of nodes that have to "ping" each other

- Detection feasible if *corrupted* nodes and *uncorrupted* nodes are connected
- Frequency settings dependent on *diameter* of connectivity graph (since determining propagation delay detection through network)

Implementation:

- <u>Cryptographic mechanism:</u>
  Entity authentication protocols
- <u>Assignment "ping" subsets:</u>
  Security manager node, key originator, self-organized, etc.

Other applications: synchronization of status information, proliferation of statistics

Conclusions & Future Directions

Technical Areas

Symmetric-key Crypto:

- Performance Crypto Mode of Operation is Right Metric, *not* Crypto Cipher

Public-key Crypto:

- Need to revisit suitability prime curves vs. binary curves, w.r.t. implementation cost, energy cost, side channel resistance, and Denial-of-Service attacks
- Public-key crypto (in HW) *is* suitable for sensor networks

Key Management:

- Need to revisit usefulness of CRLs, OCSP, certificate chains, in present-day light
- May consider separating authentication and authorization with certificates
- Key management is independent of the “color” of the key (public key, symm. key)

Trust requirements:

- Exploit “network effects”, to alleviate per-device trust requirements

Other: consider time-synch requirements, failure and out-of-synch recovery

Where Standards May Fall Short

Standards may assume capabilities that are *not* there with deployments:

- Emphasis on CRLs may kill sensor networks, which are sleepy in nature and try and minimize communication traffic density
  Standards: NIST IR 7628, Cyber Security/Smart Grid

- Sensor networks cannot easily fit crypto constructs that require AES-128 inverse block cipher implementation (e.g., ZigBee chips generally do not implement this)
  Standards: NIST SP 800-38F (key wrap)

- Key initialization should be mostly automated and consider *heterogeneous*, rather than *homogeneous*, trust environments (so as to allow mix-and-match capabilities)
  Standards: most standards leave key initialization as afterthought

Further Reading

Certificates & PKI:

1. R.L. Rivest, "Can We Eliminate Certificate Revocation Lists?," in *Financial Cryptography - FC'98*, R. Hirschfeld, Ed., Lecture Notes in Computer Science, Vol. 1465, pp. 178-183, Springer, 1998.
2. P. McDaniel, A. Rubin, "A Response to "Can We Eliminate Certificate Revocation Lists?"," in *Financial Cryptography - FC 2000*, Y. Frankel, Ed., Lecture Notes in Computer Science, Vol. 1962, pp. 245-258, Springer, 2001.
3. P. Gutmann, "PKI: It's Not Dead, Just Resting," IEEE Computer, Vol. 35, No. 8, pp. 41-48, 2002.
4. P. Gutmann, "Everything You Never Wanted to Know About PKI, but Were Forced to Find Out," University of Auckland, 2004. http://www.cs.auckland.ac.nz/~pgut001/pubs/pkitutorial.pdf
5. S. Micali, "Efficient Certificate Revocation," Technical Report TM-542b, MIT Laboratory for Computer Science, March 22, 1996.

ECC:

6. J. Fan, E. de Mulder, P. Schaumont, B. Preneel, I, Verbauwhede, "State-of-the-Art of Secure ECC Implementations: A Survey on Known Side-Channel Attacks and Countermeasures," in *3rd IEEE International Workshop on Hardware-Oriented Security and Trust - HOST 2010*, IEEE, pp. 76-87, 2010.
7. J. Taverne, A. Faz-Hernández, D.F. Aranha, F. Rodriguez-Henríquez, D. Hankerson, J. Lopez, "Software Implementation of Binary Elliptic Curves: Impact of the Carry-Less Multiplier on Scalar Multiplication", International Association for Cryptologic Research, IACR ePrint 2011-170.
8. E. Wenger, M. Hutter, "Exploring the Design Space of Prime Curves vs. Binary Field ECC-Hardware Implementations," personal communications, October 24, 2011.

Further Reading (cont'd)

ECC (cont'd):

9. D. Hein, J. Wolkerstorfer, N. Felber, "ECC Is Ready for RFID – A Proof in Silicon," in *SAC 2008*, R. Avanzi, L. Keliher, F. Sica, Eds., Lecture Notes in Computer Science, Vol. 5381, pp. 401-413, 2009.
10. "Efficient Architectures for Elliptic Cruve Cryptography Processors for RFID," in *IEEE Conference on Computer Design – ICCD 2008*, pp. 373-377, 2010.
11. T. Icart, "How to Hash Into Elliptic Curves," in *CRYPTO 2009*, S. Halevi, Ed., Lecture Notes in Computer Science, Vol. 5677, pp. 303-316, Springer, 2009.

Cryptographic Modes of Operation:

12. P. Rogaway, M. Bellare, "Encode-then-Encipher Encryption: How to Exploit Nonces or Redundancy in Plaintexts for Efficient Cryptography," in *AsiaCrypt'00*, T. Okamoto, Ed., Lecture Notes in Computer Science, Vol. 1976, Springer, 2000.
13. J.H. An, M. Bellare, "Does Encryption with Redundancy Provide Authenticity?," in *EUROCRYPT'01*, B. Pfitzmann, Ed., Lecture Notes in Computer Science, Vol. 2045, pp. 512-528, Springer, 2001.
14. NIST SP 800-38E, *Recommendation for Block Cipher Mode of Operation: The XTS-AES Mode for Confidentiality on Storage Devices*," January 2010.
15. NIST SP 800-38F, *Recommendation for Block Cipher Mode of Operation: Methods for Key Wrapping*," Draft, August 2011.

Main Current Technical Interests

Core Crypto

Efficiency Improvements:

Techniques impacting competitive positioning ECC vs. RSA (eliminating efficiency edge RSA signatures)

- 40% speed-up ECDSA signature verification (e.g., for NIST, Suite B, Brainpool curves)
- 2.4x speed-up ECDSA signature verification with ECC-based key agreement (e.g., DTLS, SSH, EAP, PGP)

Low-Power Crypto:

- Efficient crypto for highly constrained, sleepy networks
- Unbalanced and assisted computations
- Efficient key initialization for mass-produced low-cost devices (e.g., RFID, consumer goods)

Other:

- Symmetric-key crypto for bandwidth-constrained applications
- Techniques for improving security, countering attacks on ECC and other public-key schemes
- Techniques for side-channel resistance and thwarting fault attacks
- Password-based crypto schemes

Ad-hoc Sensor Networks

- Flexible configuration and trust models, minimizing need for availability infrastructure
- Semi-automatic lifecycle management, including ease of configuration, installation, and use
- Minimization of trust dependencies and need for human intervention
- Scalability, survivability, failure recovery, low impact key compromise

Ubiquitous Security

- Portable credentials (e.g., user authentication via different personal devices or via internet café)
- Security with promiscuous networks (e.g., trusted device on untrusted network, or vice-versa)