Technische Universität Braunschweig

Institute of Operating Systems and Computer Networks

µDTNSec: A Security Layer for Disruption-Tolerant Networks on Microcontrollers

Dominik Schürmann, Georg von Zengen, Marvin Priedigkeit and Lars Wolf

Med-Hoc-Net 2017

Motivation

Scenario: Huge IoT in Healthcare

- Patients
- Doctors, Nurses
- Stationary Devices (X-Ray, …)

Network Requirements for Low-Cost Deployment

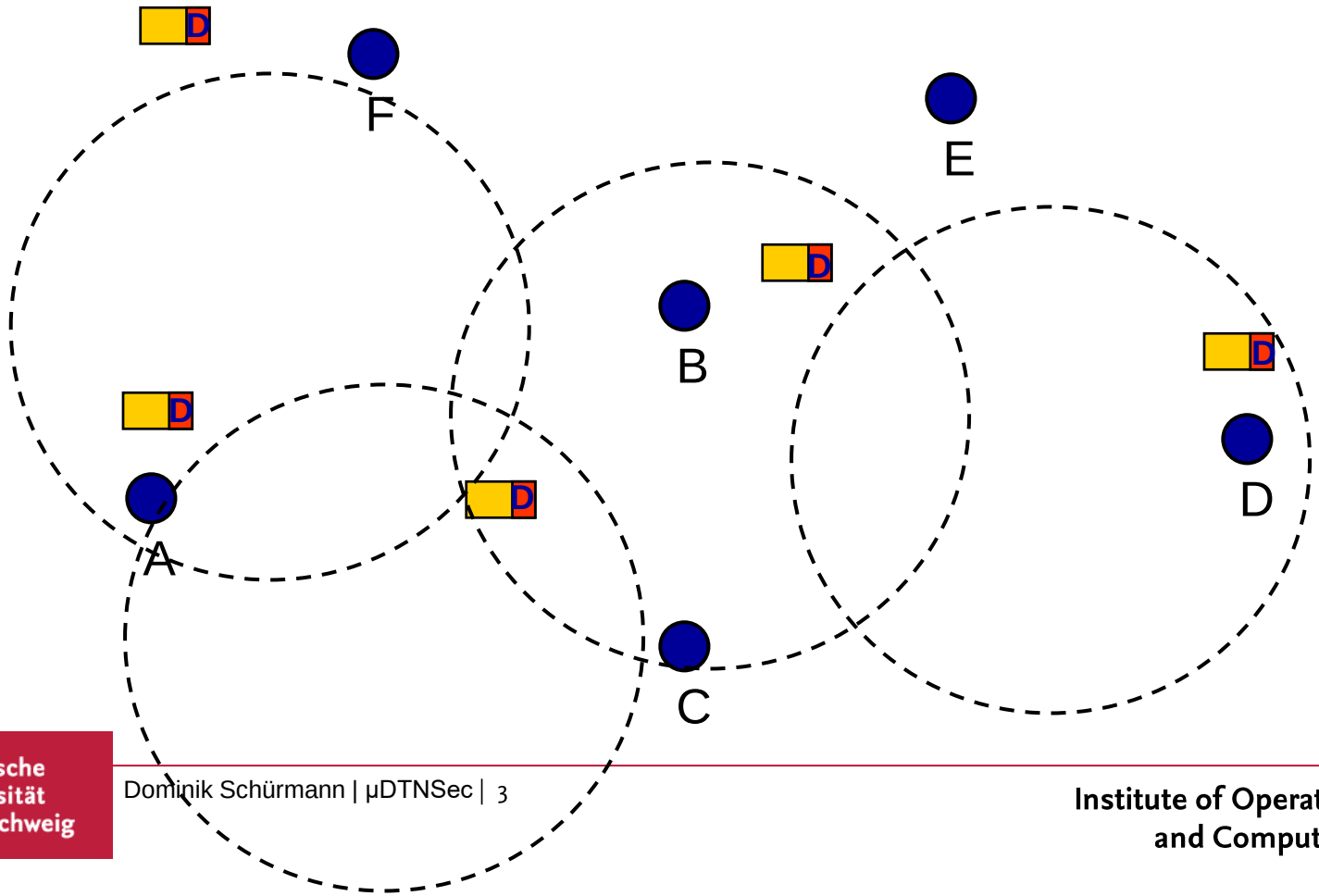
- Take advantage of node mobility (Store-Carry-Forward)
- Minimize central infrastructure

→ Traditional Internet or Ad-hoc protocols would fail

→ Deploy Delay/Disruption-Tolerant Networks (DTN)

Store-Carry-Forward

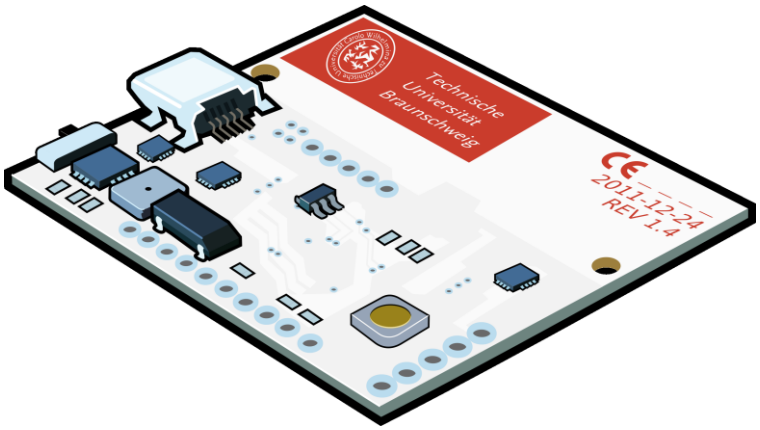
- Use disruptions and delays as an advantage



µDTN - DTN for Sensor Nodes

Implementation for

- Contiki OS (original port)
- FreeRTOS



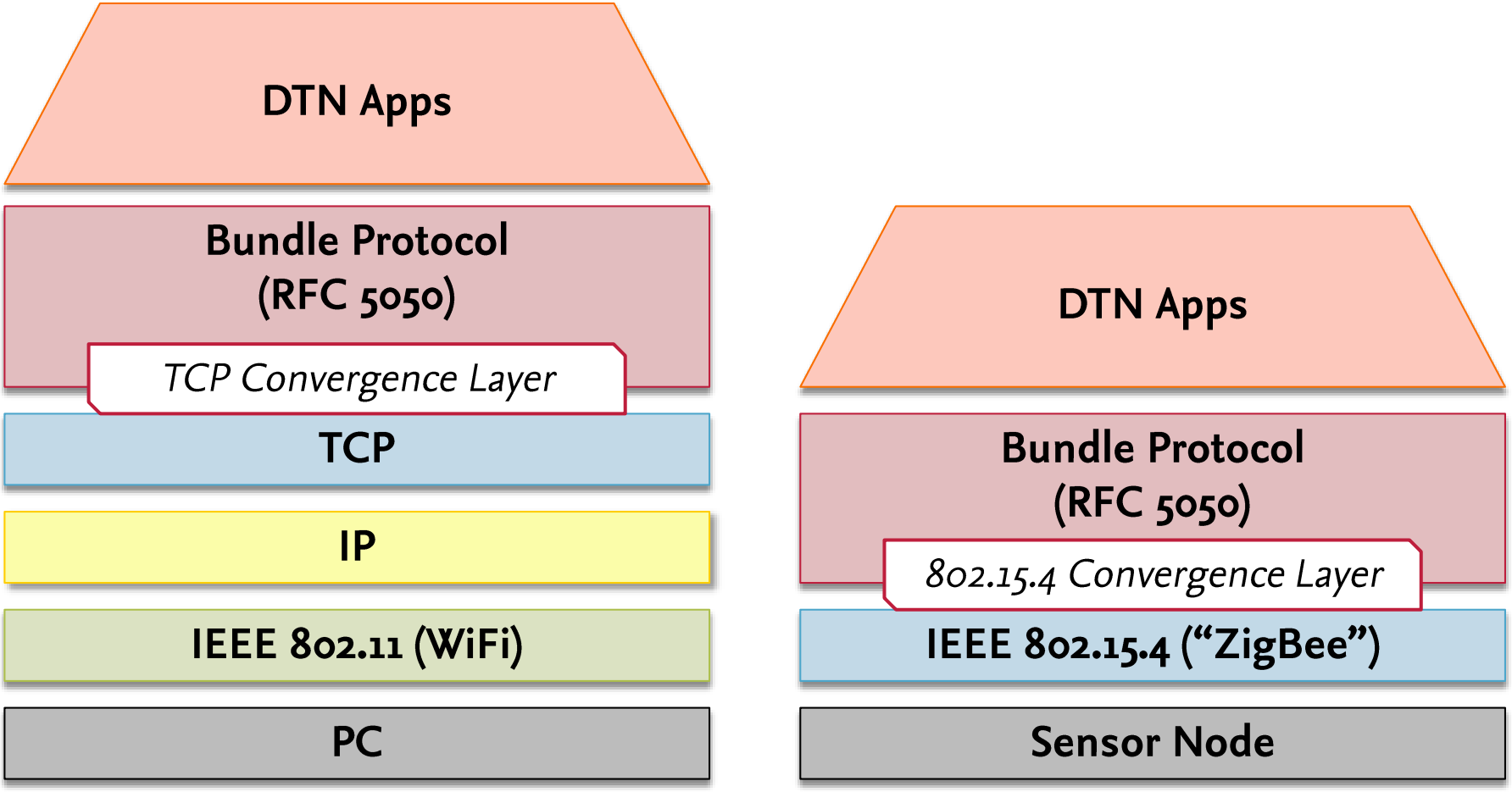
Compatible to RFC 5050

- Compressed Bundle Header Encoding (CBHE)
- Interoperable with IBR-DTN, DTN2

Located above MAC-Layer

- Contrast to most other DTN implementations
- Reduced overhead

DTN Protocol Stacks



Security Threats

Threats

- Data Modification -> Death by overdose?
- Impersonation

Attacks unbelievably easy in DTNs

More Threats

- Eavesdropping
- Man-in-the-Middle (MitM)

Traditional Security Architectures

- Example: IEEE 802.15.4 (WPAN) with single key

→ Traditional security architectures would fail on a single compromise

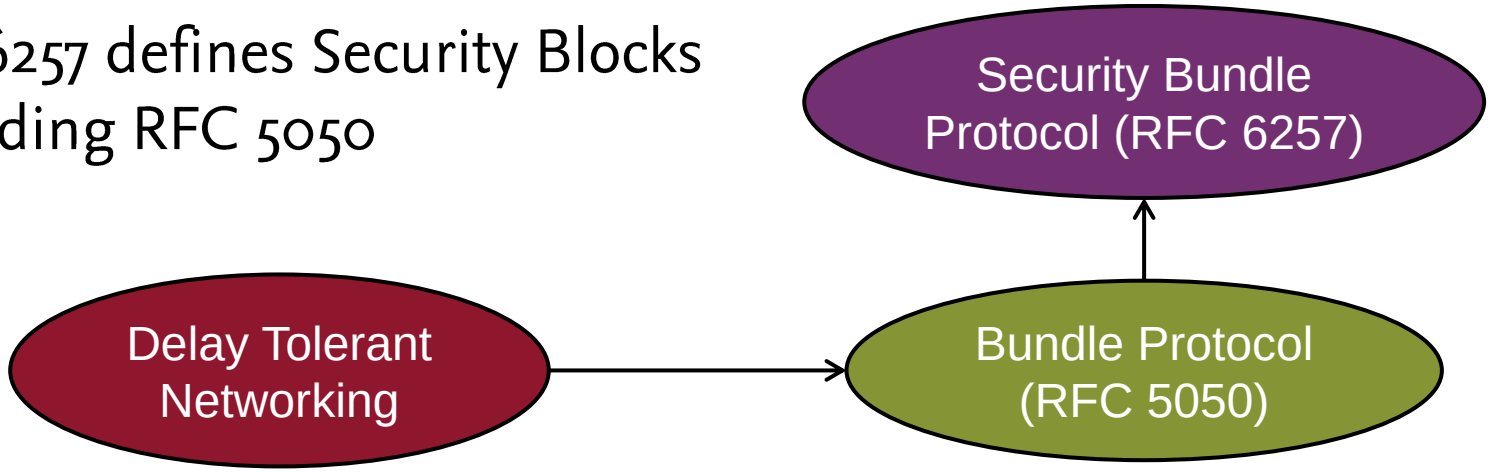
The Need for a DTN Security Layer

Goals

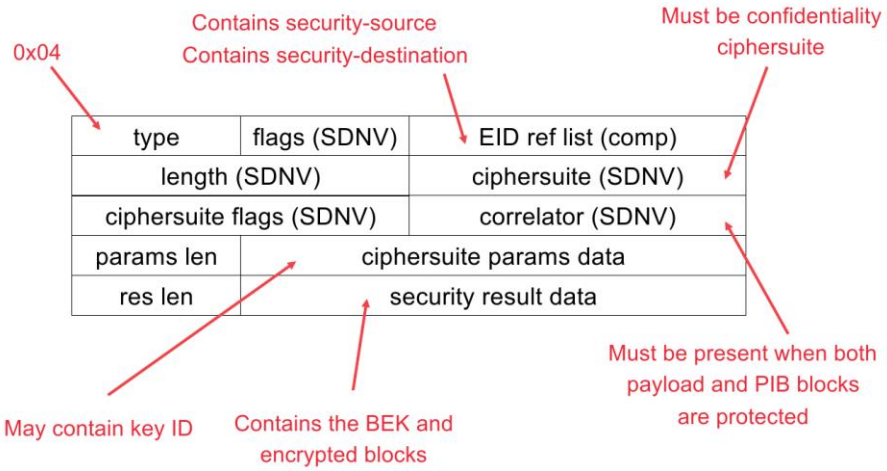
- End-to-End Security for Multihop (src -> final dst)
- Elliptic Curve Cryptography
- Use hardware AES of AT86RF23X radio

Existing Standards?

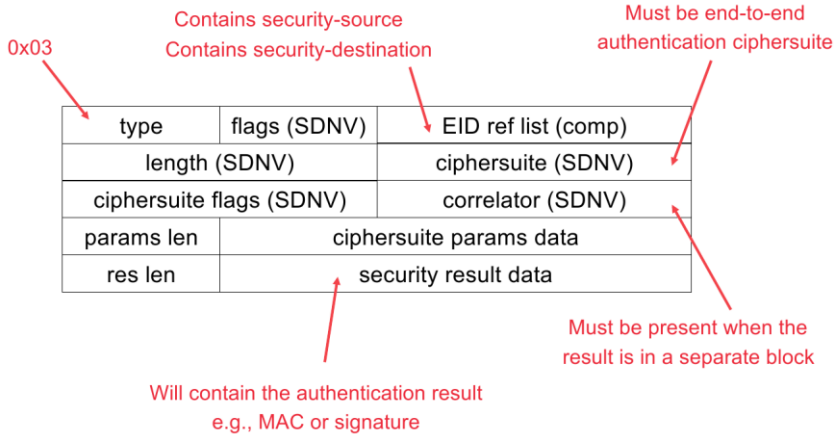
- RFC 6257 defines Security Blocks extending RFC 5050



RFC 6257: Important Blocks



Payload Confidential Block (PCB)



Payload Integrity Block (PIB)

µDTNSec Modes

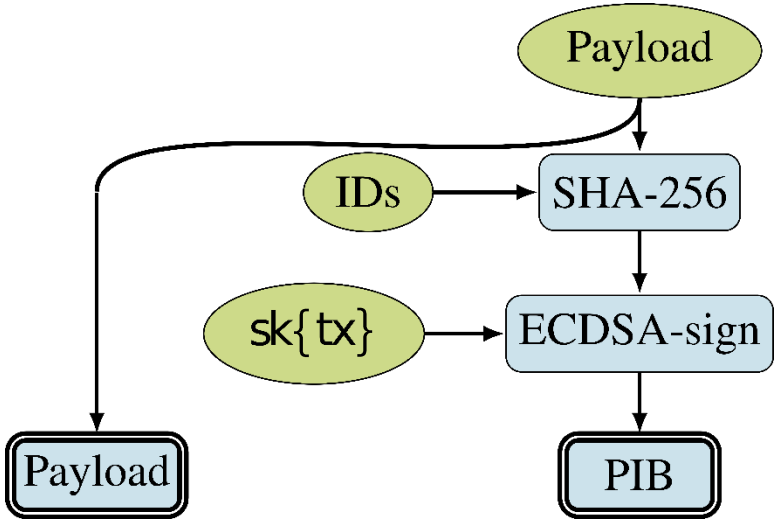
Signature Mode (PIB)

- Digital Signature with ECDSA + SHA256
- ✓ Prevents Data Manipulation and Impersonation
- ✓ Integrity/Authenticity

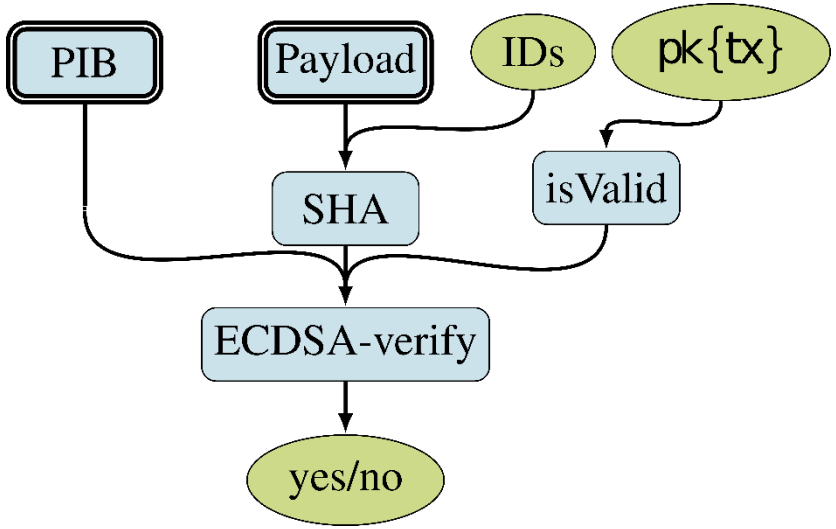
Sign-then-Encrypt Mode (PIB + PCB)

- Signature mode, then:
- Shared secret with ECDH, session key with ANSI-X9.63-KDF
- Hardware-backed AES encryption in CBC mode
- ✓ Prevents Eavesdropping, MitM attacks
- ✓ Confidentiality

µDTNSec Signature Mode



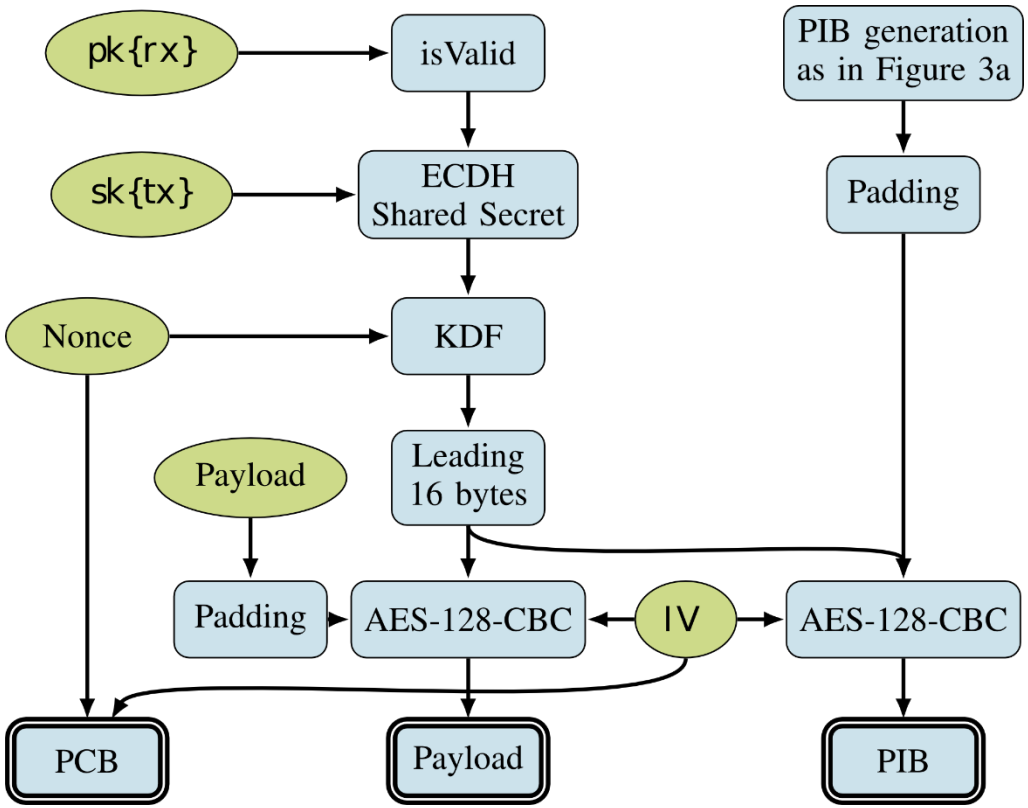
Generation



Verification

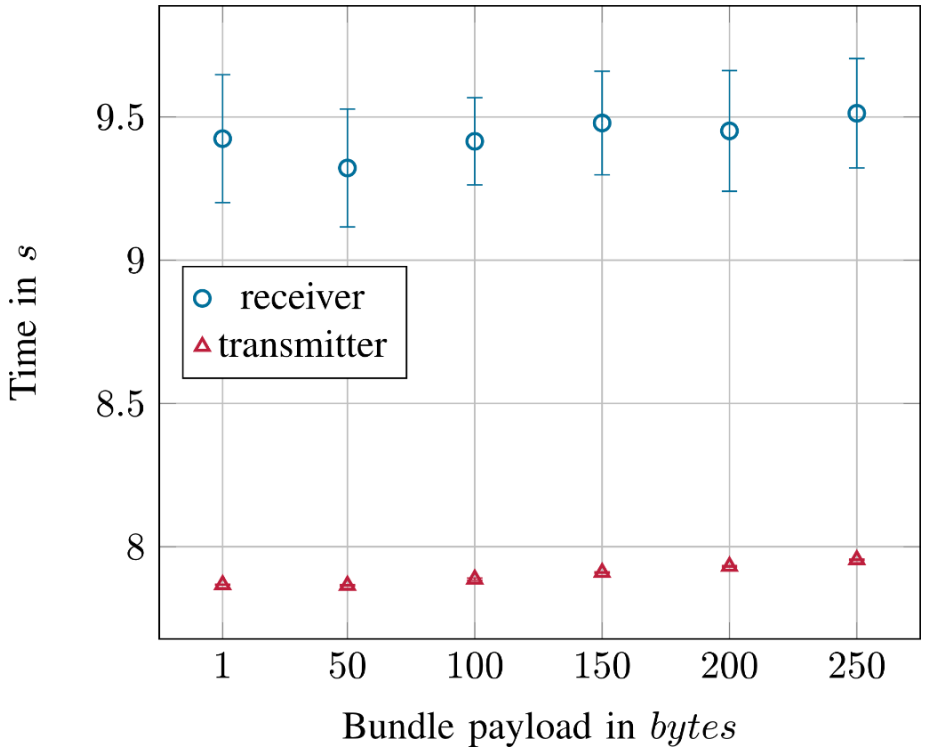
Important catch: Do not create signature over full bundle!

µDTNSec Encrypt-then-Sign

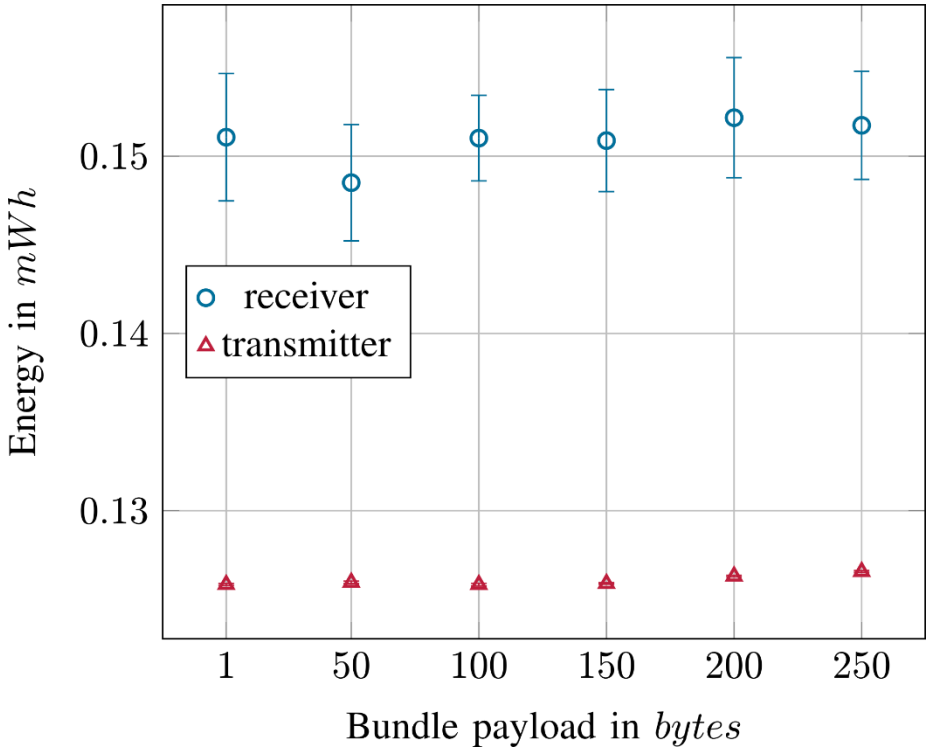


- Shared Secret between transmitter (tx) and receiver (rx)
- Check public key validity to protect against related key attacks

Signature Mode: Time and Energy

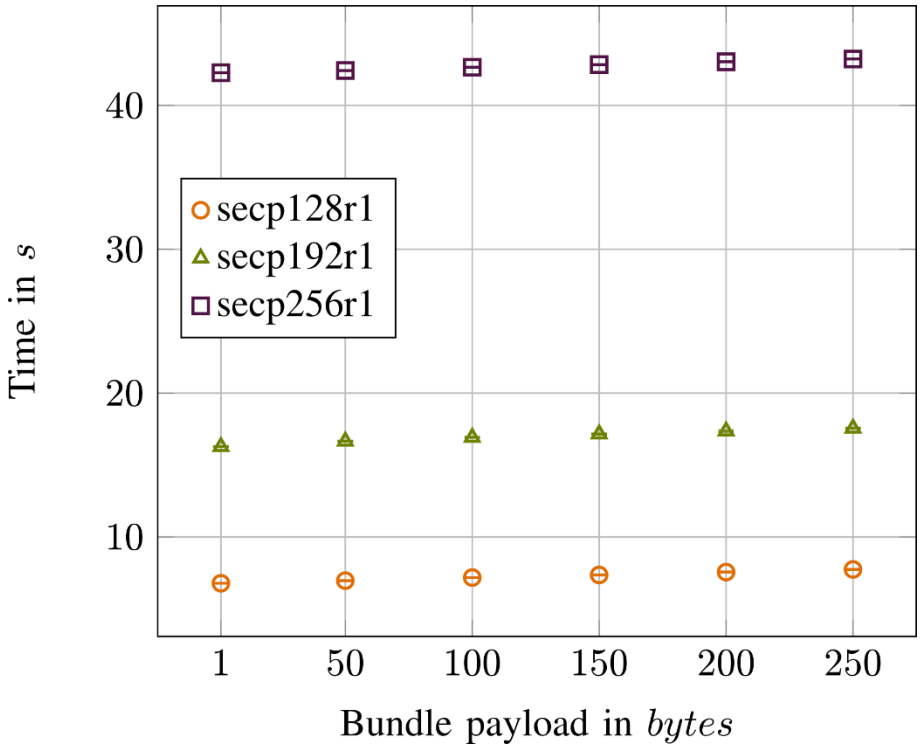


Time

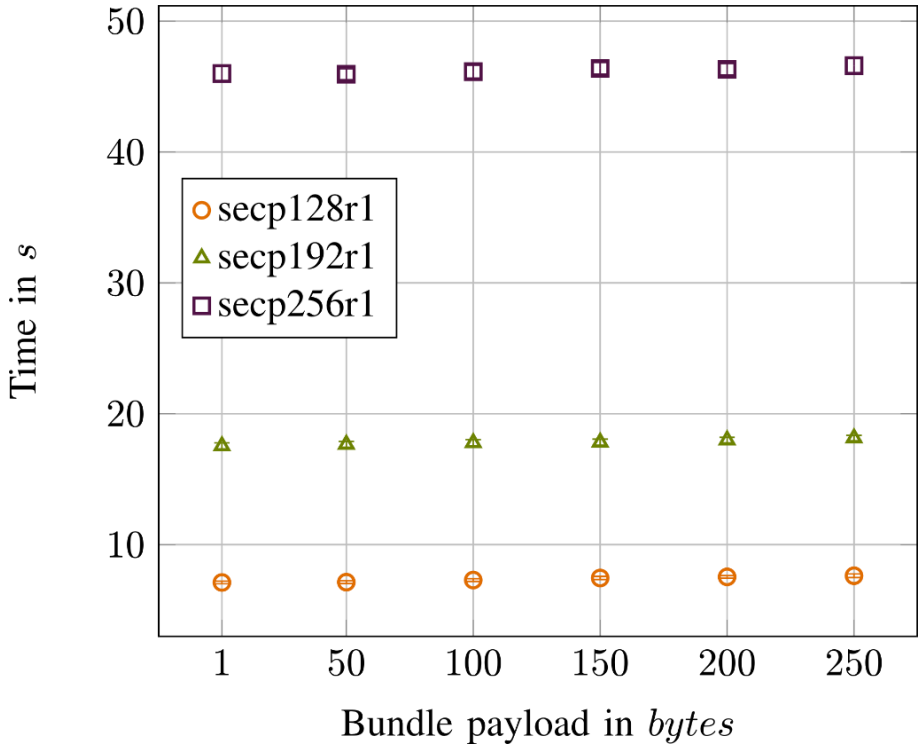


Energy

Sign-then-Encrypt Mode: Time



Transmitter



Receiver

Conclusion

- µDTNSec provides:
  End-to-End Integrity, Authenticity, Confidentiality
- Performance and Energy Measurements

- Consider requirements: Signature Mode enough?
- Transmit less often, gather more data

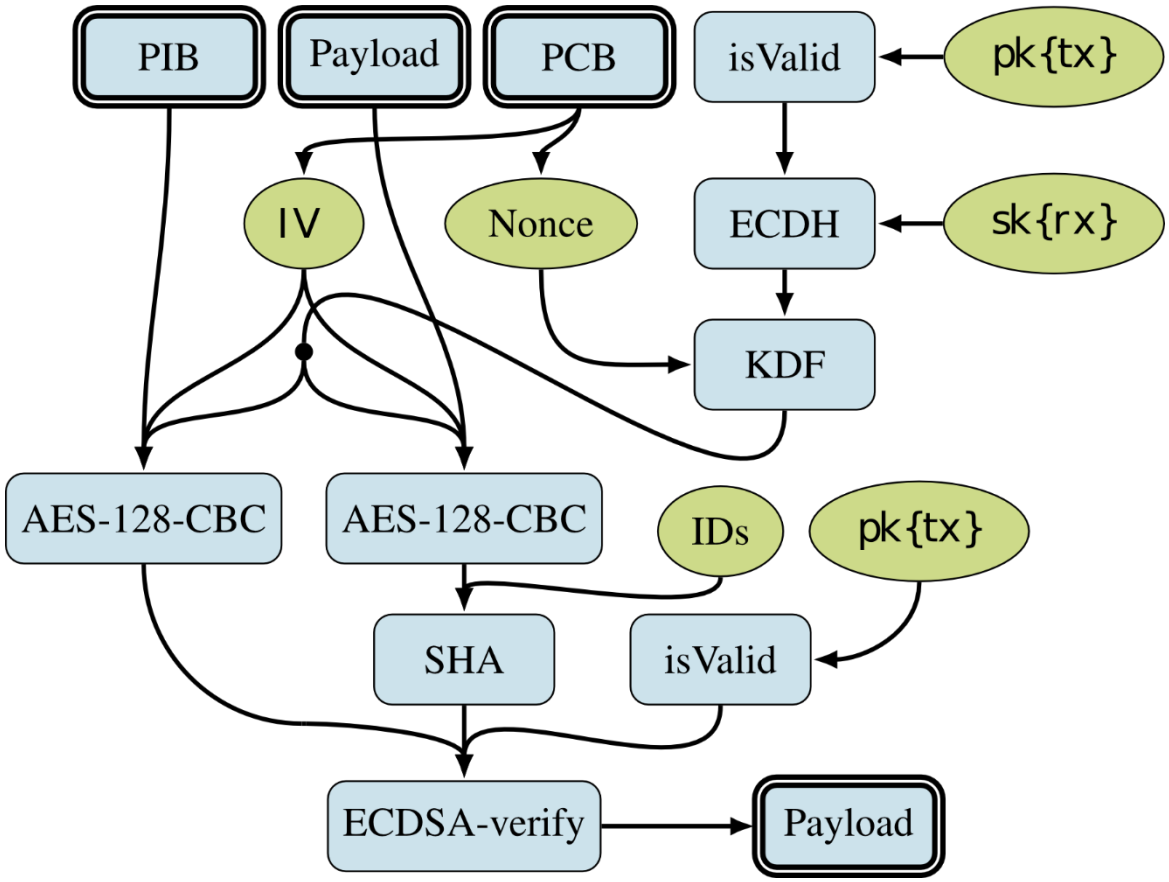
Future Work

- µDTNSec currently requires pre-deployed public keys

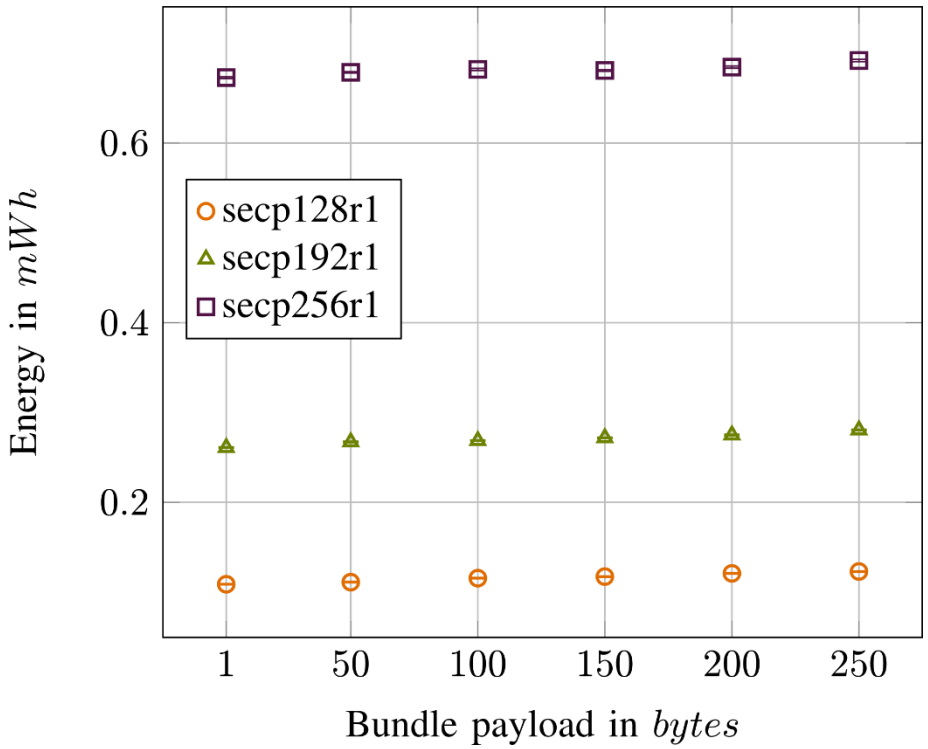
Source Code @ https://www.ibr.cs.tu-bs.de/projects/mudtn/

Backup Slides

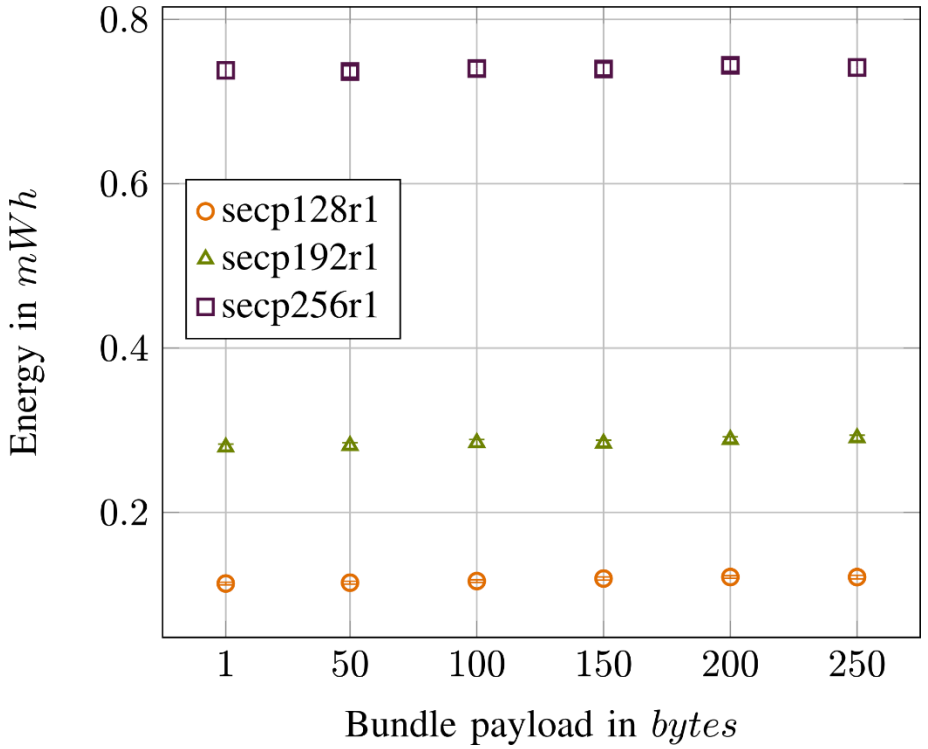
µDTNSec Encrypt-then-Sign: Decryption



Sign-then-Encrypt Mode: Energy



Transmitter



Receiver