OSCAR: Object Security Architecture for the Internet of Things*

Mališa Vučinić ★❖, Bernard Tourancheau ❖, Franck Rousseau ❖, Andrzej Duda ❖, Laurent Damon ★, and Roberto Guizzetti ★.

★ STMicroelectronics

❖ LIG - Drakkar

Crolles, April 7th 2014

life.augmented

* To appear in Proceedings of IEEE Symposium on a World of Wireless, Mobile and Multimedia Networks – WoWMoM 2014.

This presentation will make you believe public key cryptography is less costly than symmetric key (and radio communication).

Outline

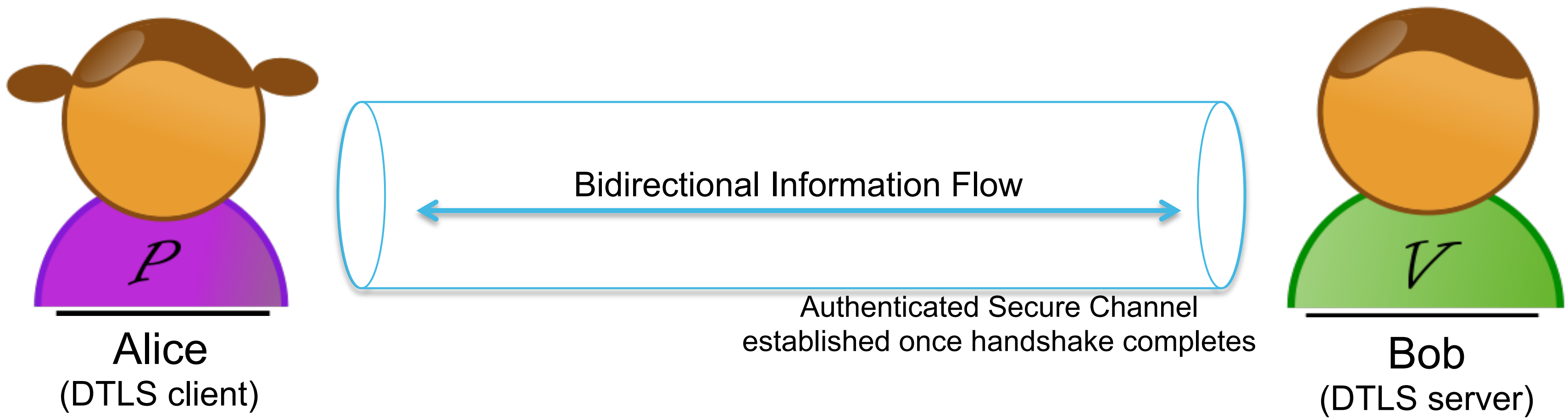
- Motivation – why not just (D)TLS
- OSCAR – concepts behind
- OSCAR – dive deep
- Implementation & Performance Evaluation
- Conclusions & Future Work

Motivation (1/6)

- Constrained Application Protocol (CoAP) in the final stage of the standardization process targeting specifically IoT applications
- CoAP main features that fulfill application requirements are [1]:
  - Group communication i.e. multicast support
  - Asynchronous message exchanges
  - Proxy and caching capabilities
  - Low overhead
  - Header mapping to HTTP
  - End-to-End Security → Solution → **DTLS**

[1] Z. Shelby, K. Hartke, and C. Bormann, “Constrained Application Protocol (CoAP) draft-ietf-core-coap-18,” IETF work in progress, 2013.

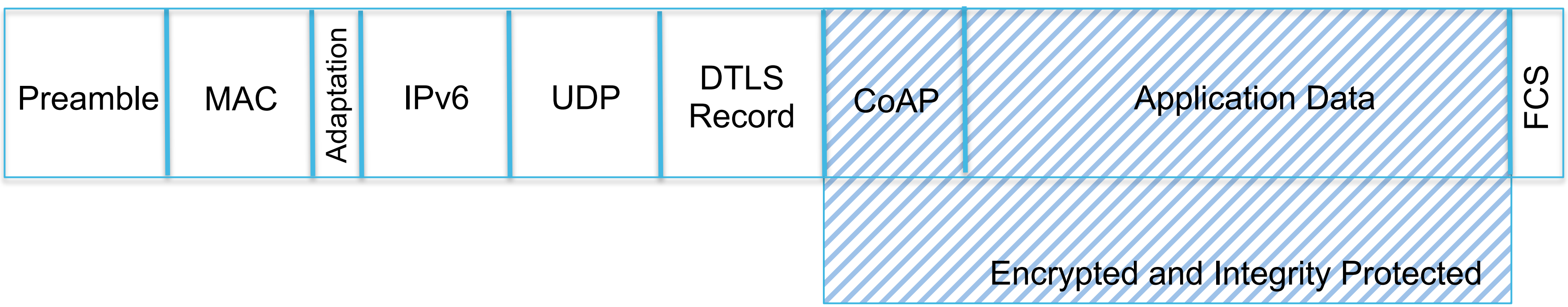
DTLS Recap (1/3)



Motivation (2/6)

- CoAP + DTLS features (CoAPs):
  - Group communication i.e. multicast support ✗
  - Asynchronous message exchanges ±
  - Proxy and caching capabilities ✗
  - Low overhead
  - Header mapping to HTTP
  - End-to-End Security

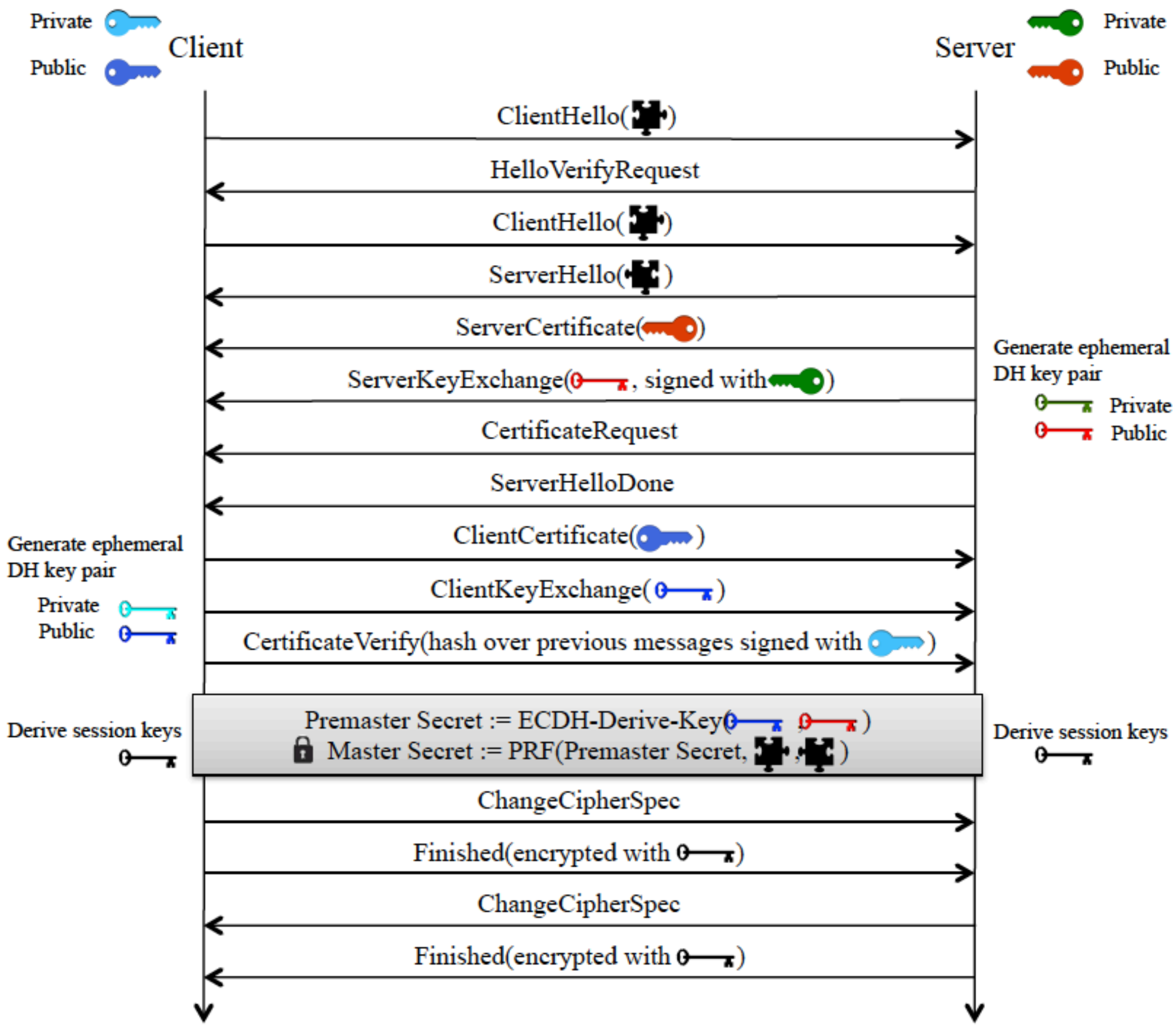
DTLS Recap (2/3)



Motivation (4/6)

- CoAP + DTLS features (CoAPs):
  - Group communication i.e. multicast support ✗
  - Asynchronous message exchanges ±
  - Proxy and caching capabilities ✗
  - Low overhead
  - Header mapping to HTTP ✗
  - End-to-End Security

DTLS Recap (3/3)



[2] H. Shafagh, 'Leveraging Public-key-based Authentication for the Internet of Things', Master's thesis, RWTH Aachen University, 2013.

Motivation (5/6)

- CoAP + DTLS features (CoAPs):
  - Group communication i.e. multicast support ✗
  - Asynchronous message exchanges ±
  - Proxy and caching capabilities ✗
  - Low overhead ✗
  - Header mapping to HTTP ✗
  - End-to-End Security ✓

Motivation (6/6)

- Security that can't support **basic** application requirements is of no use
- Fundamental design choices of CoAP and DTLS are incompatible
  - (D)TLS targets connection oriented **point-to-point** application flows (Voice over IP, some online games)
  - Only basic request-response mechanism of CoAP could be regarded as connection-oriented. What about:
    - Asynchronous notification and observation of resources with dynamic group membership
    - Caching and integration with the Cloud
- TCP and its three-way handshake (syn, syn-ack, ack) were ruled out from LLNs due to "terrible performance"
  - 3 RTT and 10-15 packet DTLS handshake, with completion time from several seconds to **one minute** (depending on the duty-cycle) **?**
  - Statements like "it [handshake] is performed only once during the initialization phase and/or later (rarely) for re-handshake" [3] should make us think as if we may as well hard-code logical topologies and interactions

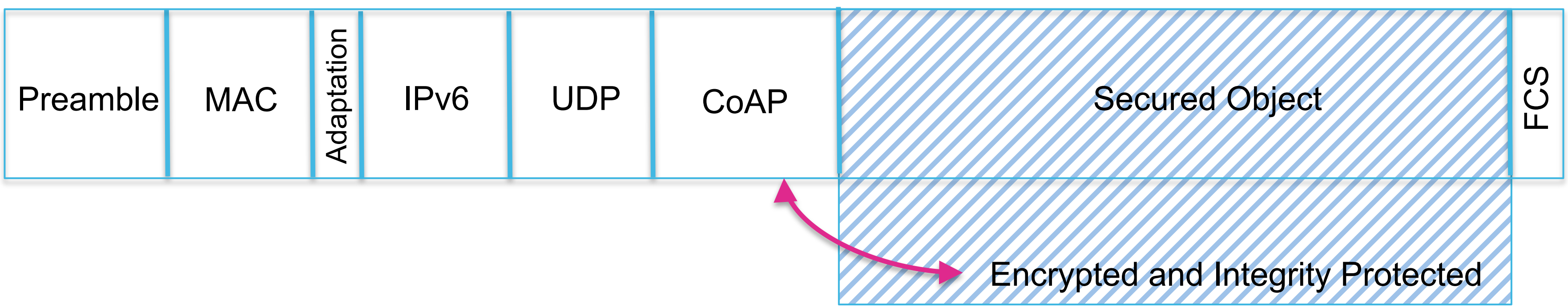
[3] S. Raza, H. Shafagh, K. Hewage, R. Hummen, and T. Voigt, "Lithe: Lightweight Secure CoAP for the Internet of Things," Sensors Journal, IEEE, vol. 13, no. 10, pp. 3711–3720, 2013.

OSCAR – concepts behind (1/3)

- **Requirement 1:** Make security features compatible with application requirements, not vice-versa
- **Requirement 2:** Allow E2E security in presence of statefull gateways that do not allow direct communication from the outside and the WSN
- **Requirement 3:** Backwards compatibility with plain DTLS approach, as standardized in CORE WG, to support existing deployments
- **Requirement 4:** High practical value targeting IETF efforts on End-to-End security and Authorization in constrained environments (CORE, DICE and ACE WGs)
- **Requirement 5:** Minimal energy consumption to allow most energy-stringent devices, like GreenNet nodes

OSCAR – concepts behind (2/3)

- **Idea 1:** A stateless security architecture
  - Allows caching, eases group communication and asynchronous exchanges
  - Solution: Application-level security i.e. Object security (CMS, JOSE)
  - Protect from communication-related replay attacks by binding object-security encryption keys with underlying CoAP duplicate detection mechanism



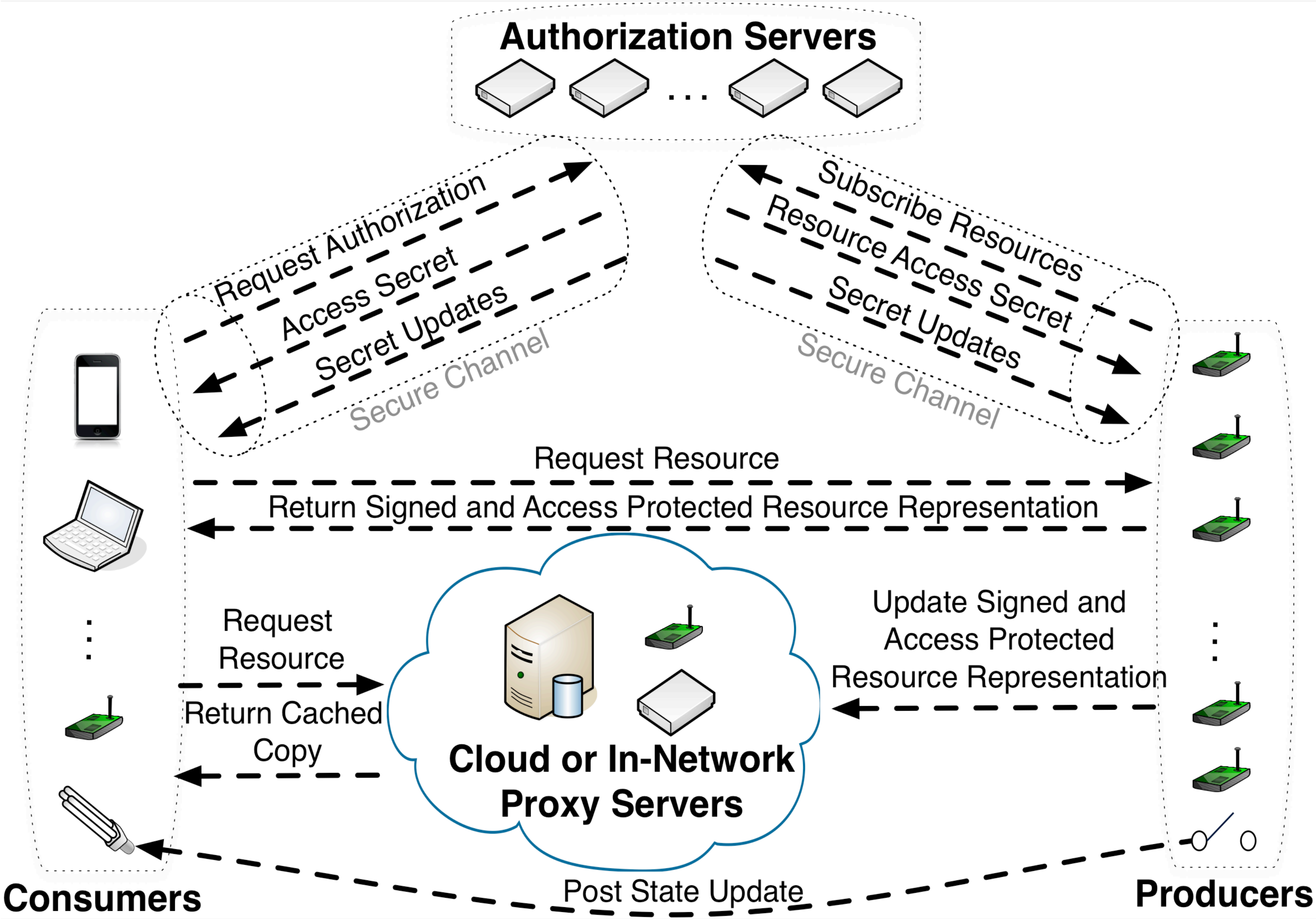
OSCAR – concepts behind (3/3)

- **Idea 2:** Move the burden of security handshake away from constrained servers
  - Introduce a **semi**-trusted, non-constrained third party that will do the hard work
  - Constrained servers respond with secured objects (resource representations) regardless of the identity of the client
- **Idea 3:** Jointly approach problems of End-to-End security and Authorization
  - Split confidentiality and authenticity trust domains
  - Confidentiality used to provide access-control for group members
  - Authenticity strongly tied to the originator of the information (individual sensor)

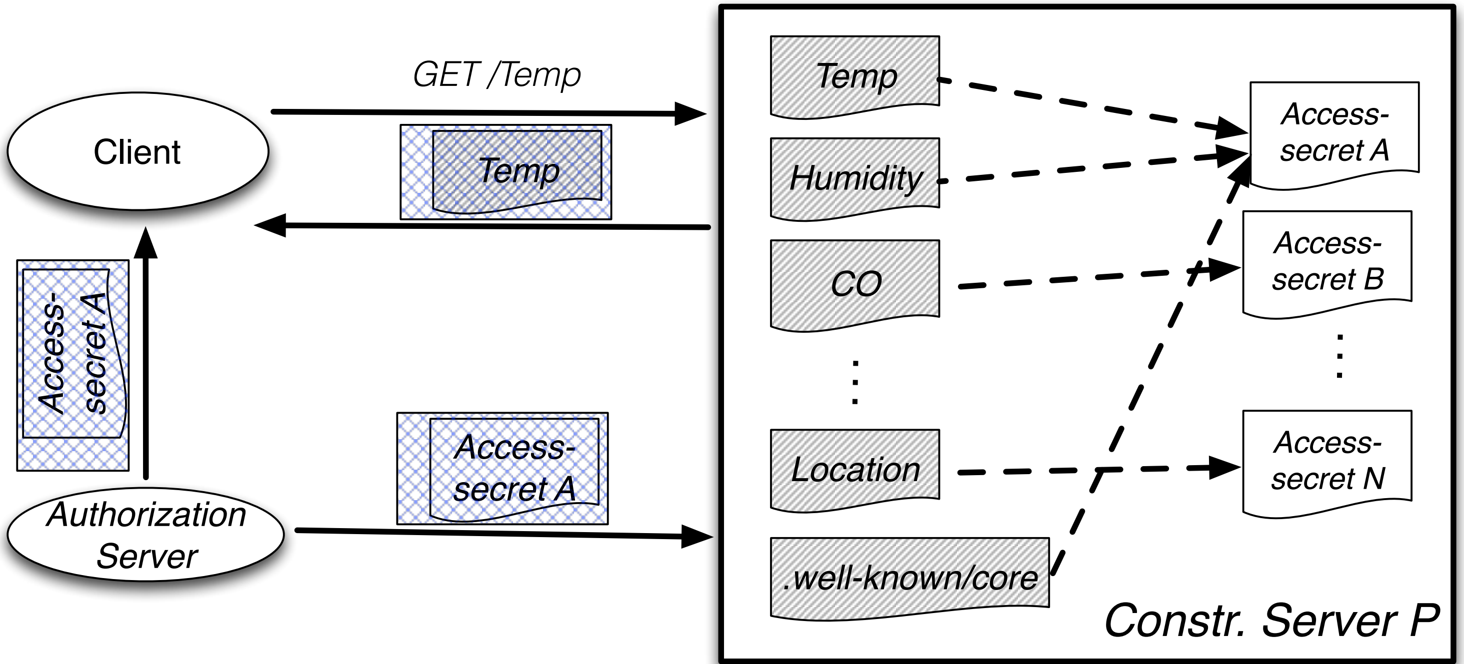
OSCAR – dive deep (1/4)

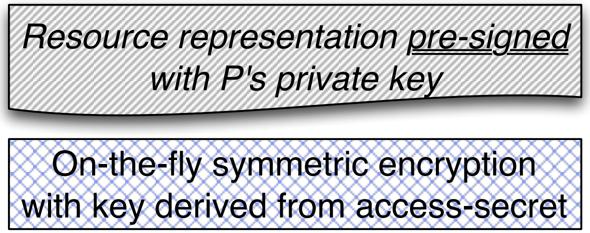
- We use the Producer-Consumer model to provide security
  - Producers: sensors, smart-meters, motion detectors, switches, …
  - Consumers: actuators, mobile devices, collection centers, human users, …
- Producers' main task is to generate information and to secure it **independently** of possibly many consumers
  - We **decouple** the public-key cryptographic overhead from network communication on the producer side
    - Results in functional simplicity of producers (constrained nodes)
  - Producers update **secured** resources as they are observed in the environment
    - This allows lots of application-specific optimizations to reduce the cryptographic overhead
  - Producers respond to all requests with access-protected resource representations (symmetric encryption)
  - Main processing burden is shifted away from producers (constrained servers)
- Consumers fetch the information either from intermediate proxies, the Cloud, by direct CoAP request/response interaction or they are asynchronously notified of changes (CoAP observe option)

OSCAR – dive deep (2/4)



OSCAR – dive deep (3/4)





OSCAR – dive deep (4/4)

- In summary:
  - We heavily use digital signatures to provide authenticity of information tied to individual sensors (source authentication)
    - Surprisingly good performance results in comparison with DTLS-only approach
    - Work on-going to support use-cases where this is not practical
  - We use confidentiality to provide capability-based access-control by symmetric encryption
    - Protection against communication related replay-attacks by binding the actual encryption key to the duplicate detection mechanism of CoAP
  - Authorization Server(s) in charge of authentication and distribution of appropriate access-secrets
  - Implicit compatibility with multicast and caching
  - DTLS used for communication with Authorization Servers and to enable backwards compatibility

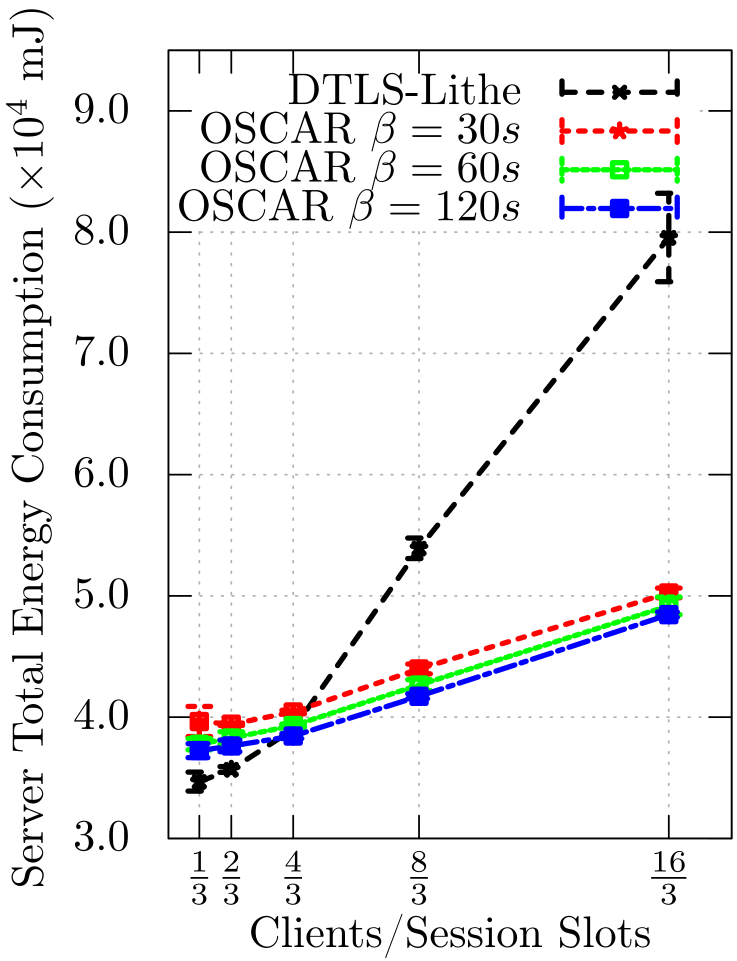
CoAP + OSCAR

- CoAP + OSCAR features:
  - Group communication i.e. multicast support ✓
  - Asynchronous message exchanges ✓
  - Proxy and caching capabilities ✓
  - Low overhead ±
  - Header mapping to HTTP ✓
  - End-to-End Security ✓
  - Authorization and Access Control ✓

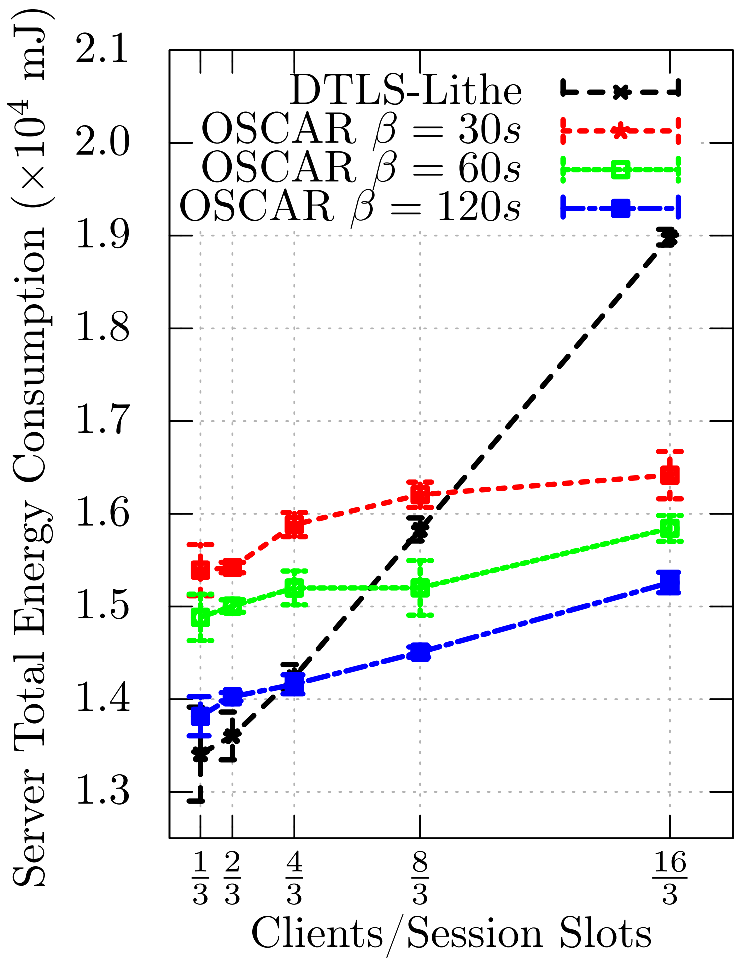
Implementation & Performance Evaluation

- Developed object-security library for Contiki
  - Supports encrypted, signed and encrypted/signed object types.
  - Coupled with CoAP to provide cipher negotiation capabilities and replay protection
- Evaluation
  - Two hardware platforms at 21.3 MHz:
    - WiSMote - 16-bit MSP430, 16K RAM, CC2520 radio transceiver
    - GreenNet tags - STM32L, 32K RAM, RF200W radio transceiver
  - We study scalability as a function of number of clients per server
    - Number of simultaneous DTLS sessions is limited due to memory constraints of nodes
    - With a simple application for evaluation purposes, we could fit up to 3 simultaneous sessions for WiSMote. Same number used with GreenNet to have comparable results.
  - Pre-shared key based cipher suite for DTLS using only **symmetric** key operations
  - TinyECC library for OSCAR using secp160r1 elliptic curve for signing
  - Typical 6LoWPAN stack (CoAP, UDP, IPv6, 6LoWPAN, 802.15.4)
- Methodology
  - Each point averaged over five 3-hour runs and plotted with 95% conf. intervals

Server-side Total Energy Consumption

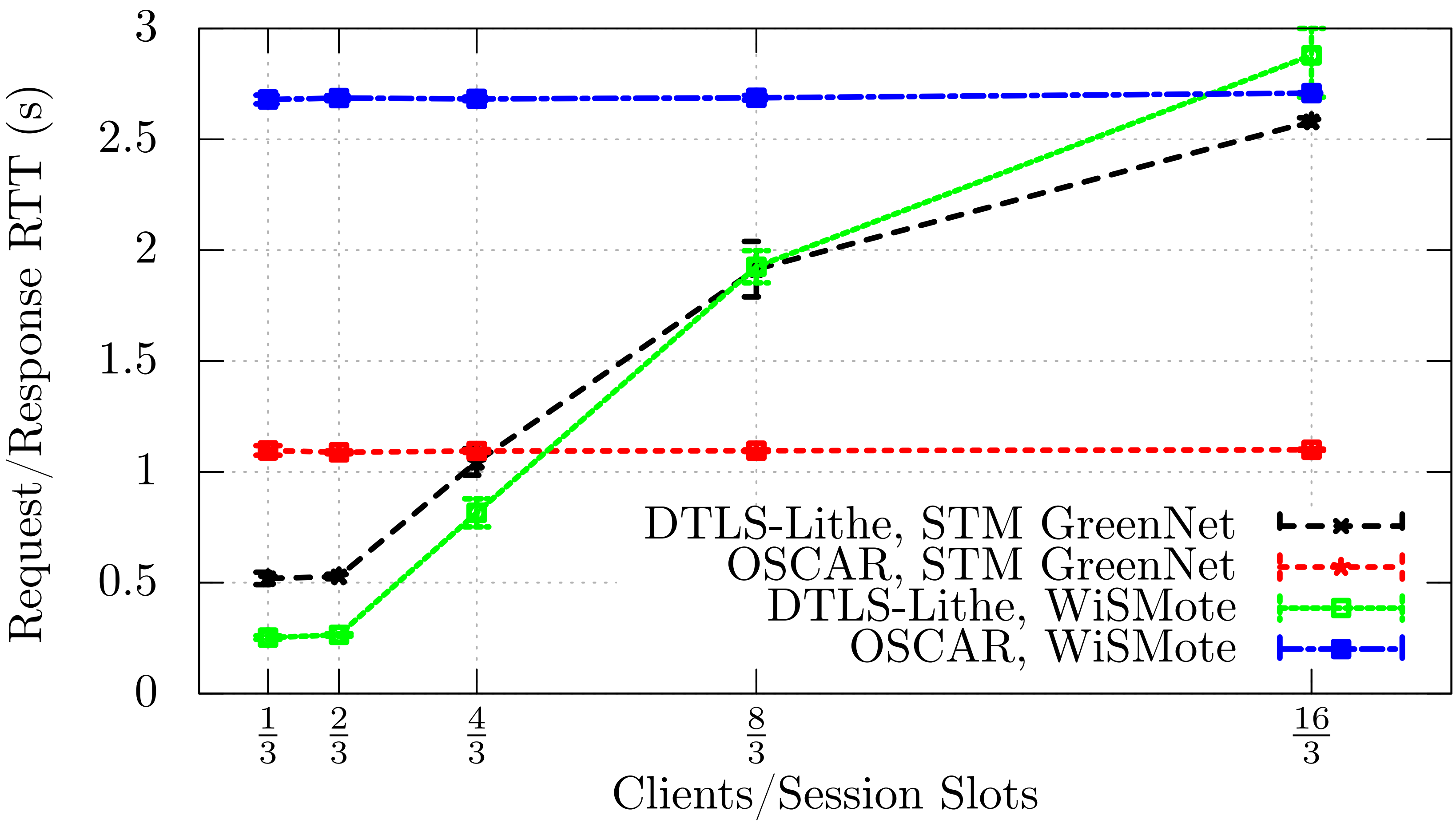


WiSMote

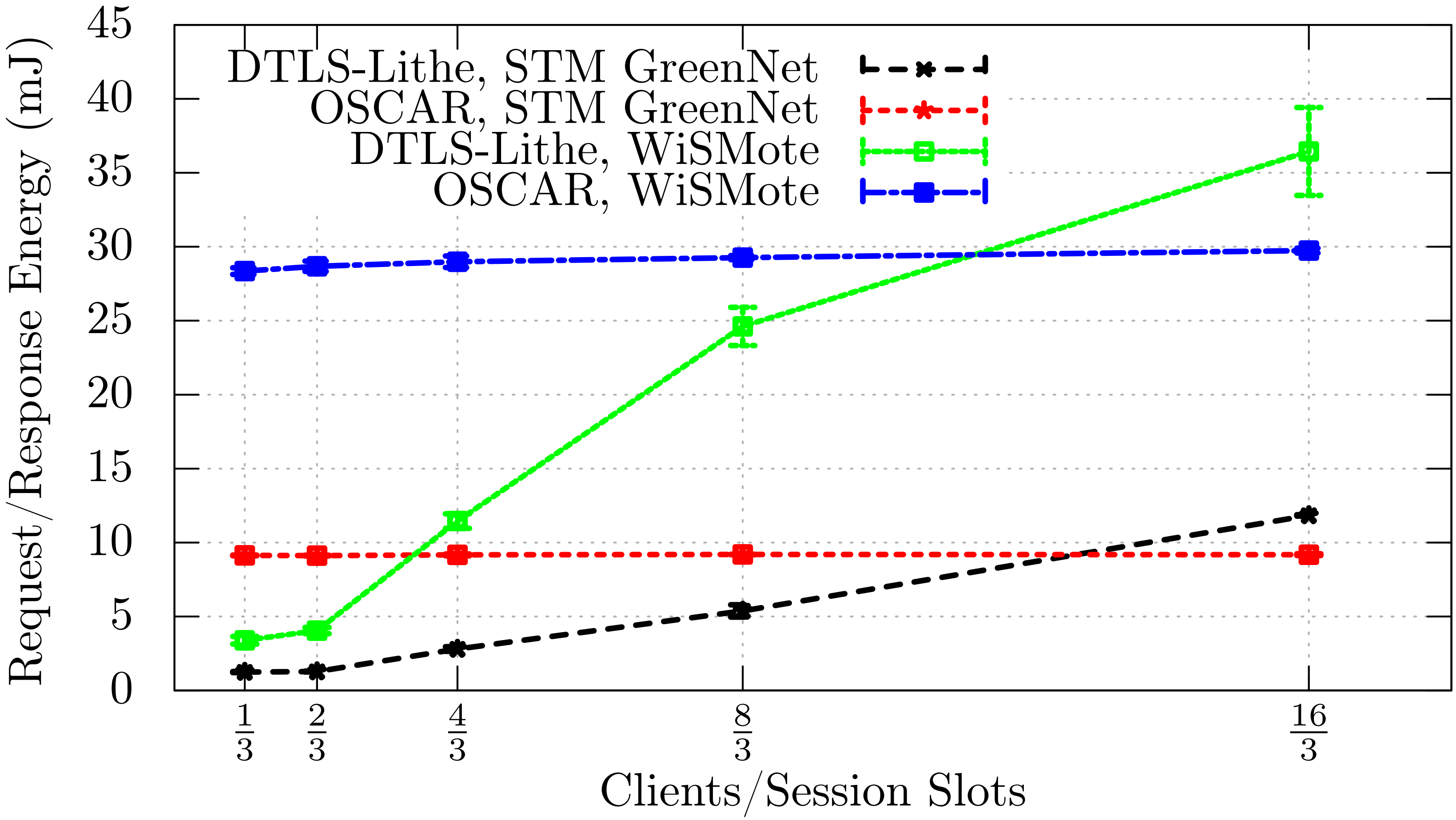


ST GreenNet

Client-side Request/Response Delays



Client-side Request/Response Energy



Conclusions & Future Work

- Established E2E security and authorization framework that actually supports application requirements
- Can provide E2E security even in presence of statefull gateways
- Particularly useful for use-cases where high number of clients per-constrained-server is expected
  - Smart city a very good example
- Future work required
  - Use-cases that require streaming where constant digital signing is unfeasible
  - Key management and authorization policies

Hvala lijepa!* Questions?

*Merci bien!

Backup

Implementation & Performance Evaluation