TerraSwarm

A Toolkit for Construction of Authorization Service Infrastructure for the Internet of Things (IoT)

Hokeun Kim[1], *Eunsuk Kang*[1], *Edward A. Lee*[1], *David Broman*[2]

[1]University of California, Berkeley

[2]KTH Royal Institute of Technology

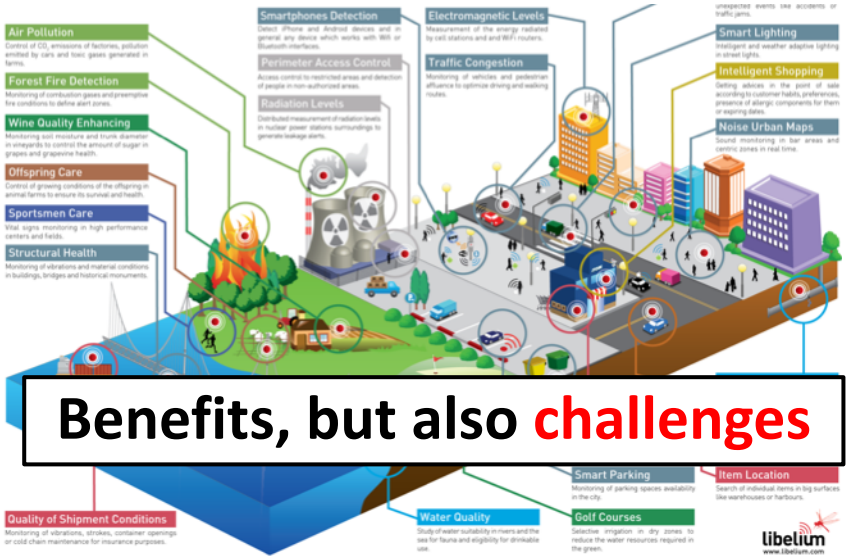
IoTDI 2017, Pittsburgh, PA

April 19, 2017

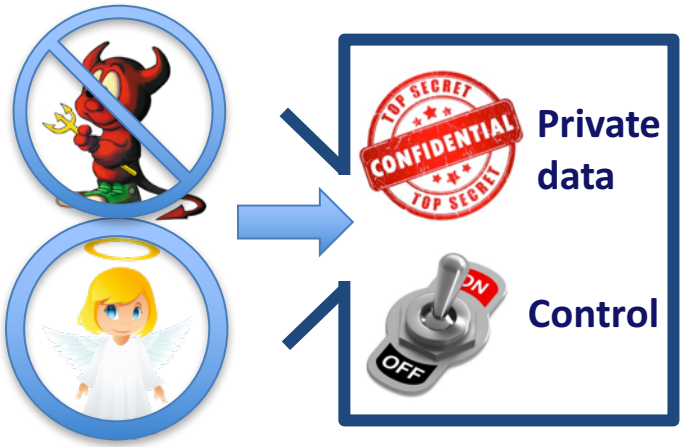
Sponsored by the TerraSwarm Research Center, one of six centers administered by the STARnet phase of the Focus Center Research Program (FCRP) a Semiconductor Research Corporation program sponsored by MARCO and DARPA.

Overview – IoT & Authorization

- **Internet of Things**



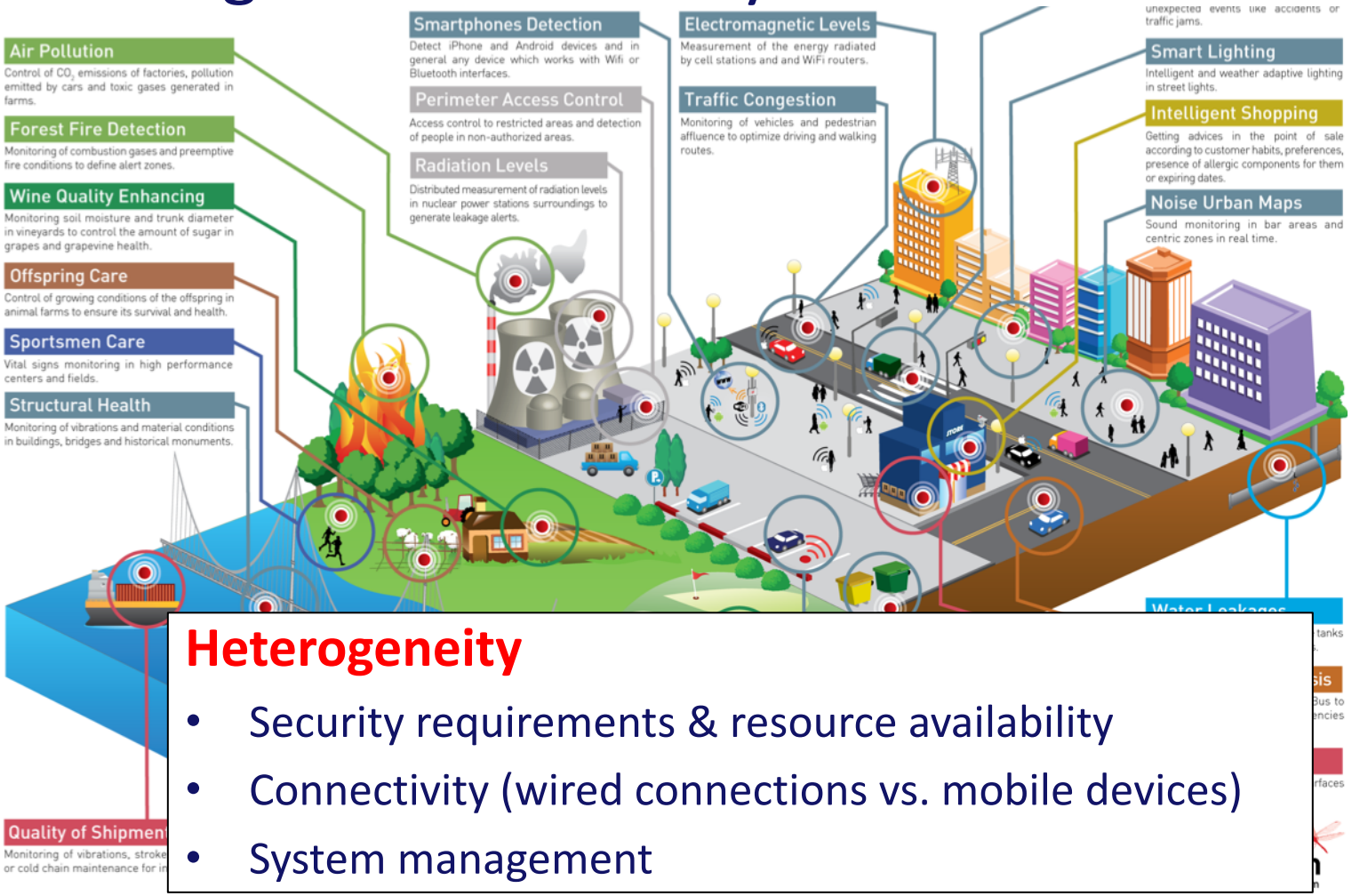
- **Authorization (access control)**
  - Critical for computer security



- **Existing security solutions?**
  - May work well for some parts of the IoT, but not for the entire IoT!
- **Proposed approach – SST**
  - SST: Secure Swarm Toolkit
  - An open-source toolkit for building authorization infrastructure for the IoT
  - To address IoT security challenges

Motivation

- Challenges in IoT security[1]



Heterogeneity

- Security requirements & resource availability
- Connectivity (wired connections vs. mobile devices)
- System management

[1] Singh *et al.*, 2016. "Twenty Security Considerations for Cloud-Supported Internet of Things"

Motivation (cont'd)

- Challenges in IoT security[1]

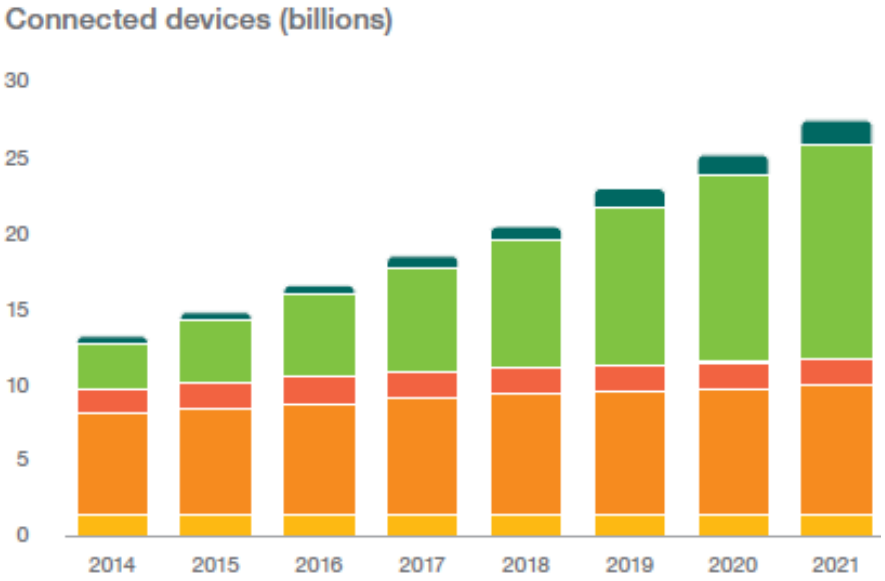
Operation in an open (or hostile) Environment

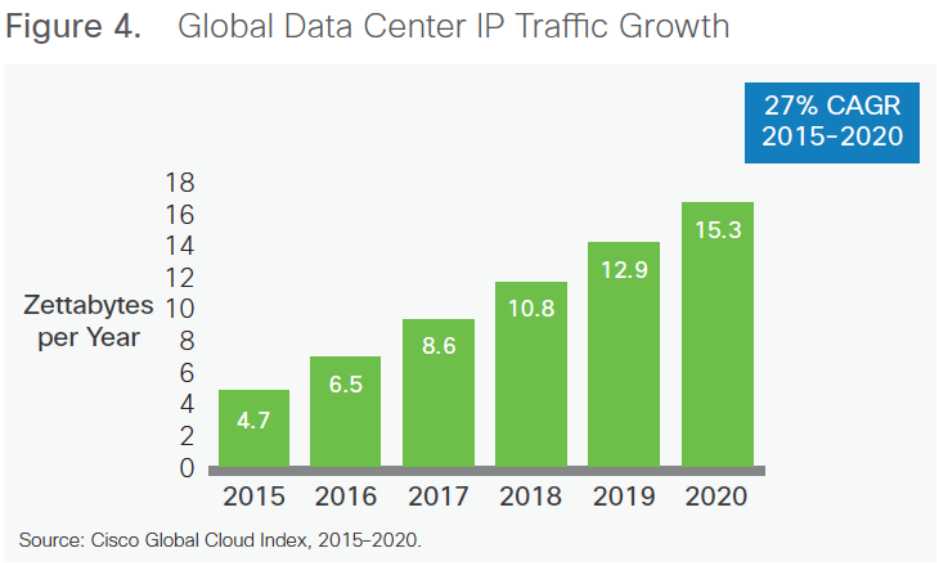
- Physical access & wireless access to IoT devices
- Higher risk of being compromised
- Must be able to revoke access of compromised IoT devices

[1] Singh *et al.*, 2016. "Twenty Security Considerations for Cloud-Supported Internet of Things"

Motivation (cont'd)

- Challenges in IoT security[1]





Sources: "Ericsson Mobility Report", June 2016 / "Cisco Global Cloud Index: Forecast and Methodology,2015–2020", Published in 2016

Scalability

- 28 billion connected devices in 2021
- 15.3 ZB data traffic in 2020
  - 1 ZB (Zetta byte) = 10^9 TB (Terra bytes)

[1] Singh *et al.*, 2016. "Twenty Security Considerations for Cloud-Supported Internet of Things"

Background: Authorization & IoT

- Authorization
  - Access control
    - "Can I enter the EECS building?"
  - Allowing/denying access to resources
  - Revoking access (e.g., lost ID card)
- Authentication
  - Identifying someone/something
    - "Member of EECS?"
  - Essential for authorization

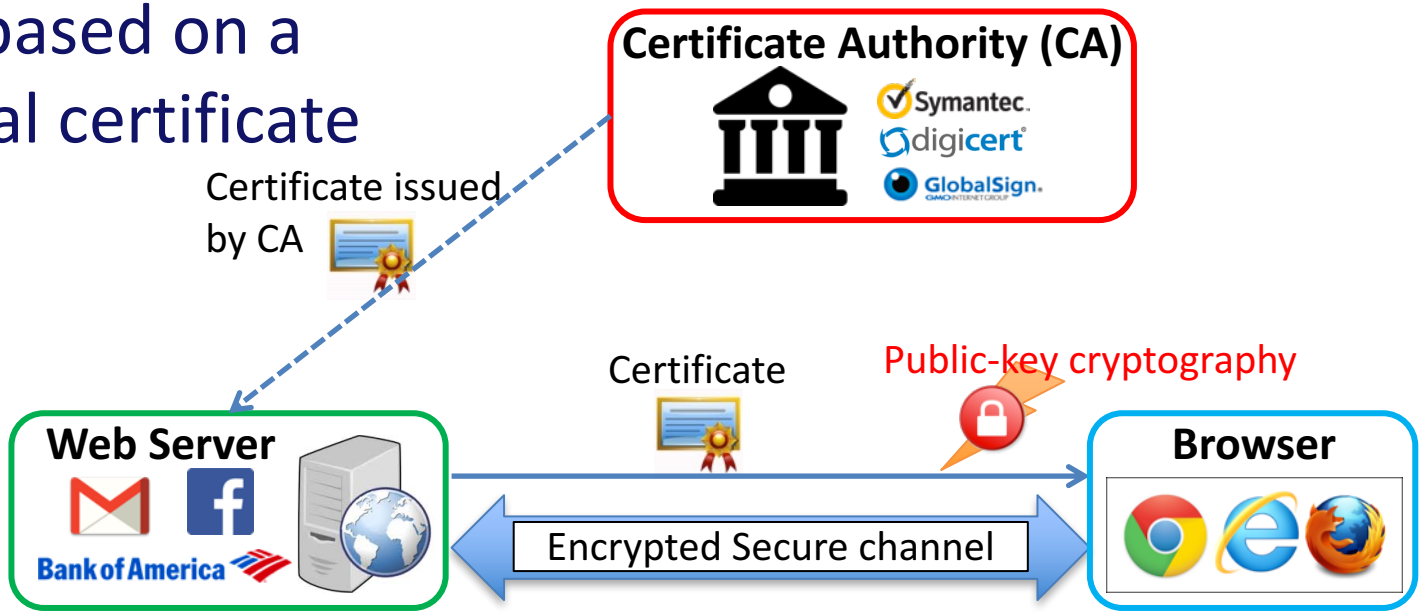
Background (cont'd)

- Many IoT platforms use TLS (or DTLS[2]) for authentication/authorization
  - E.g., Amazon AWS IoT, OpenIoT[3], OSCAR[4], etc.
- TLS (Transport Layer Security, also called SSL/TLS)
  - Underlying security protocol for HTTPS
  - Widely used, very successful for web

[1] Variant of TLS over UDP, 2012 "Datagram Transport Layer Security Version 1.2. RFC 6347"
[2] John Soldatos et al., 2015. "OpenIoT: Open Source Internet-of-Things in the Cloud"
[3] Vucinic et al., 2015. "OSCAR: Object security architecture for the Internet of Things"

Background (cont'd)

- TLS based on a digital certificate

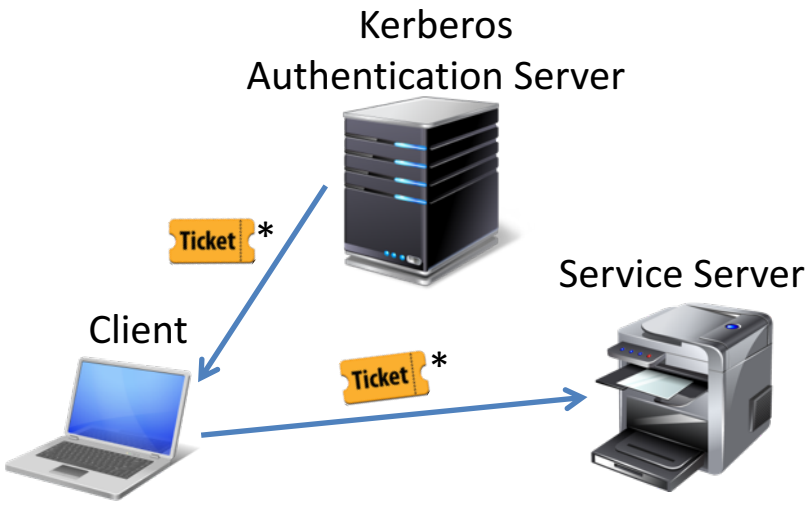


- Challenges with using TLS for the entire IoT
  - Energy overhead of public-key crypto & certificates
  - Scalability (managing certificates for ~28 billion devices)
  - Revocation of certificates can be problematic[1,2]
  - Limited support for one-to-many communication

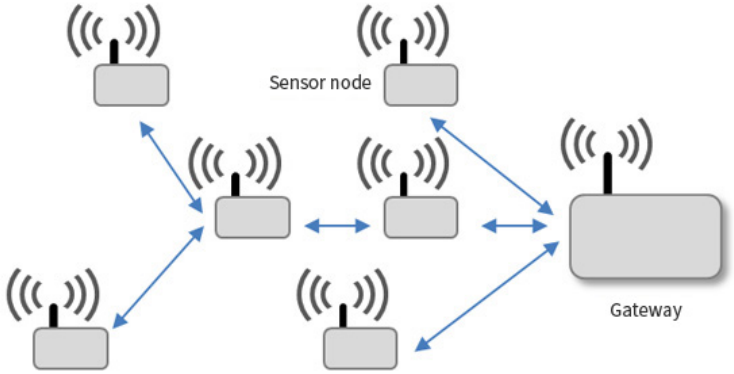
[1] Mutton, "Certificate revocation: Why browsers remain affected by Heartbleed", Netcraft, April, 2014
[2] Duncan, "How certificate revocation (doesn't) work in practice", Netcraft, May, 2013

Background (cont'd)

- Challenges with applying other security solutions



*Ticket: temporary token for accessing service



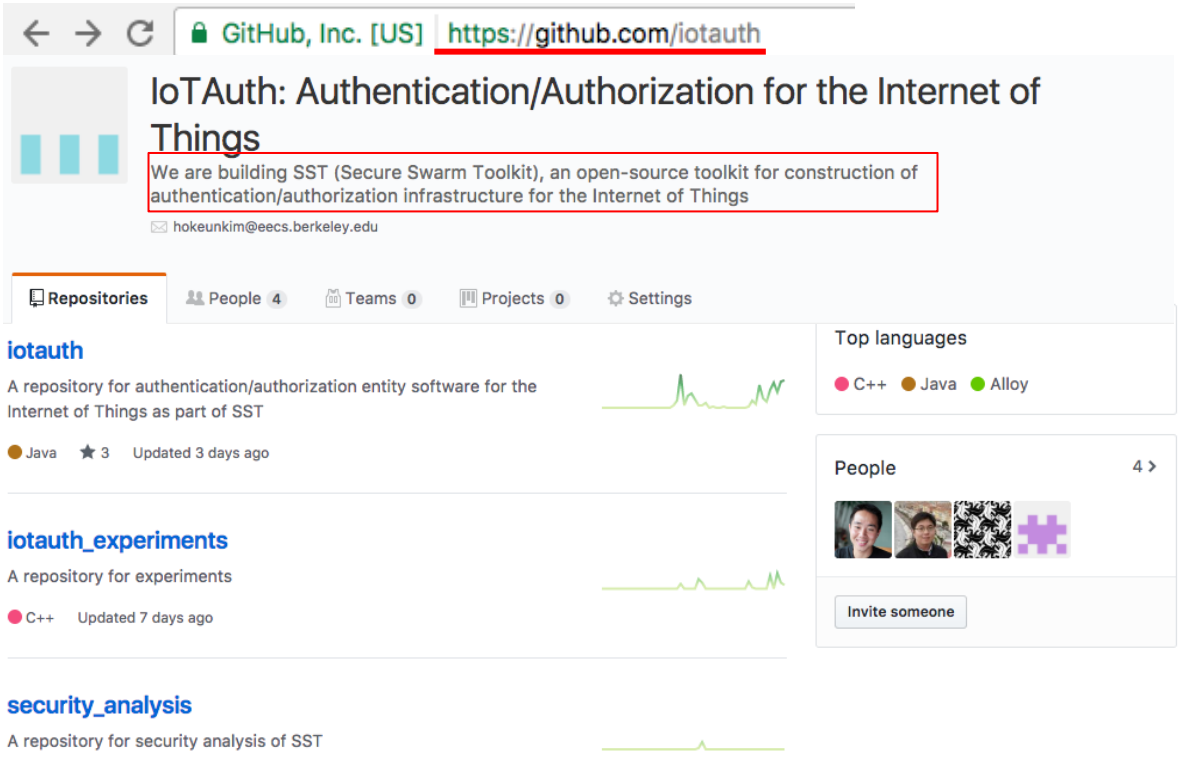
Source: http://www.yuden.co.jp/ut/solutions/wsn/

- Kerberos [1]
  - Advantages for access revocation
  - Requires stable connection
  - Centralized architecture
- Security solutions for "Things"
  - E.g., WSN, MANET or swarm devices
  - Assume homogeneous environments
  - Not designed for Internet scale[2]

[1] C. Neuman et al., 2005. "The Kerberos Network Authentication Service (V5)". RFC 4120
[2] Alcaraz et al., 2010. "Wireless sensor networks and the internet of things: Do we need a complete integration?"

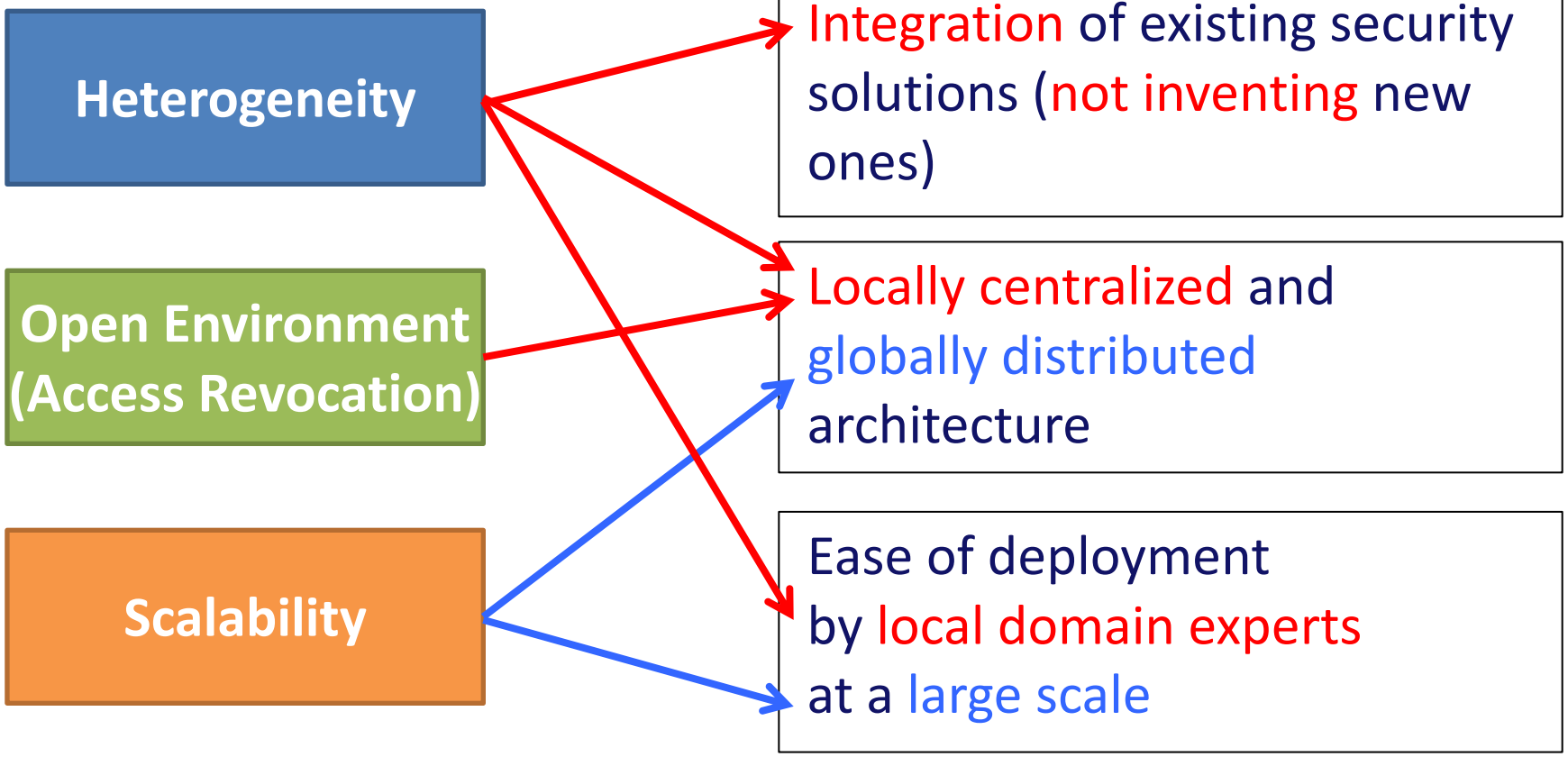
Proposed Approach

- **SST – Secure Swarm Toolkit**
  - An open-source toolkit for authentication/authorization of the IoT (available on https://github.com/iotauth)



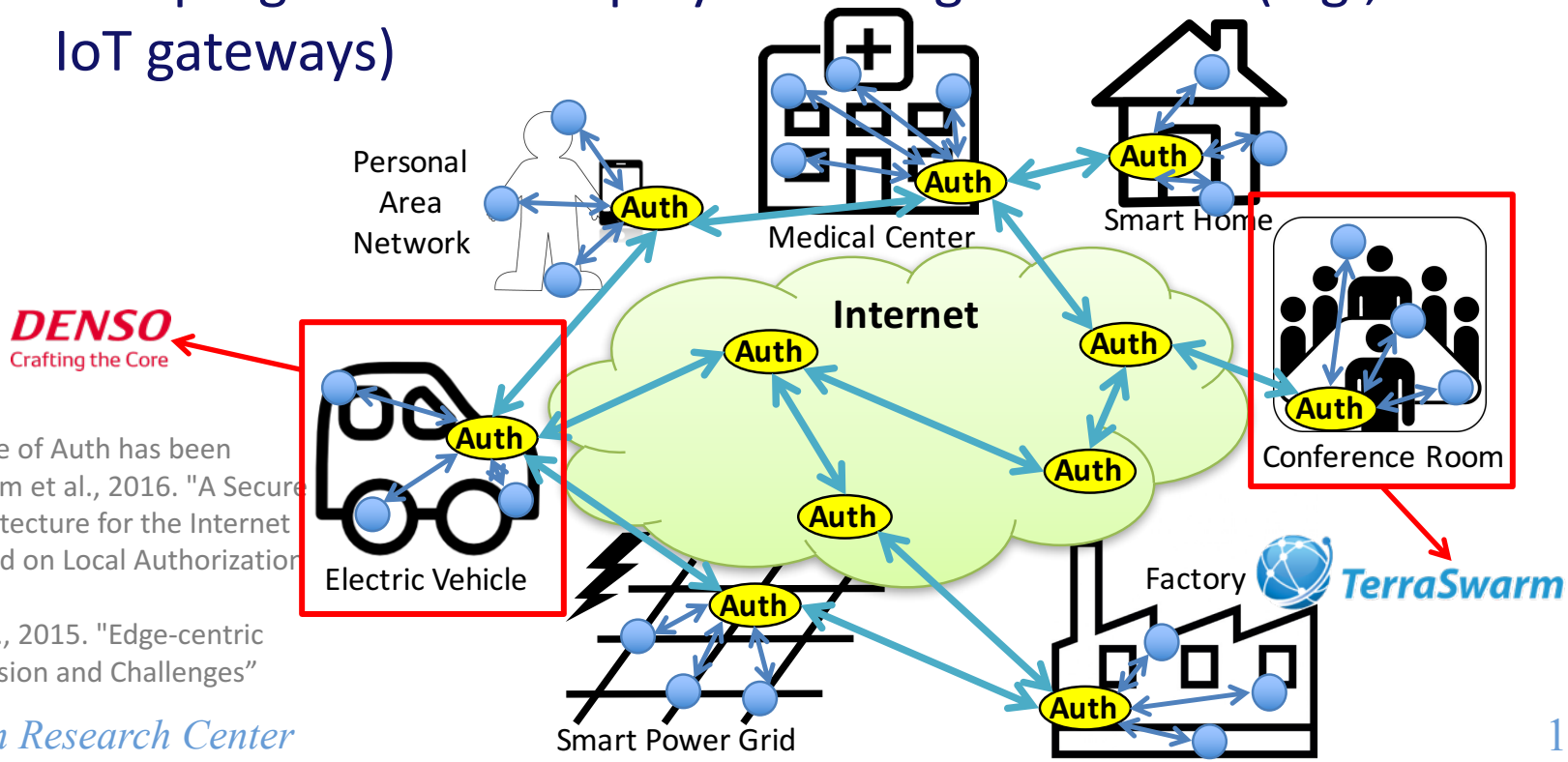
Proposed Approach (Cont'd)

- Specific goals of SST



SST's Design and Implementation

- Auth[1]
  - Locally centralized, globally distributed **auth**entication/**auth**orization entity (software)
  - Java program to be deployed on edge devices[2] (e.g., Intel IoT gateways)

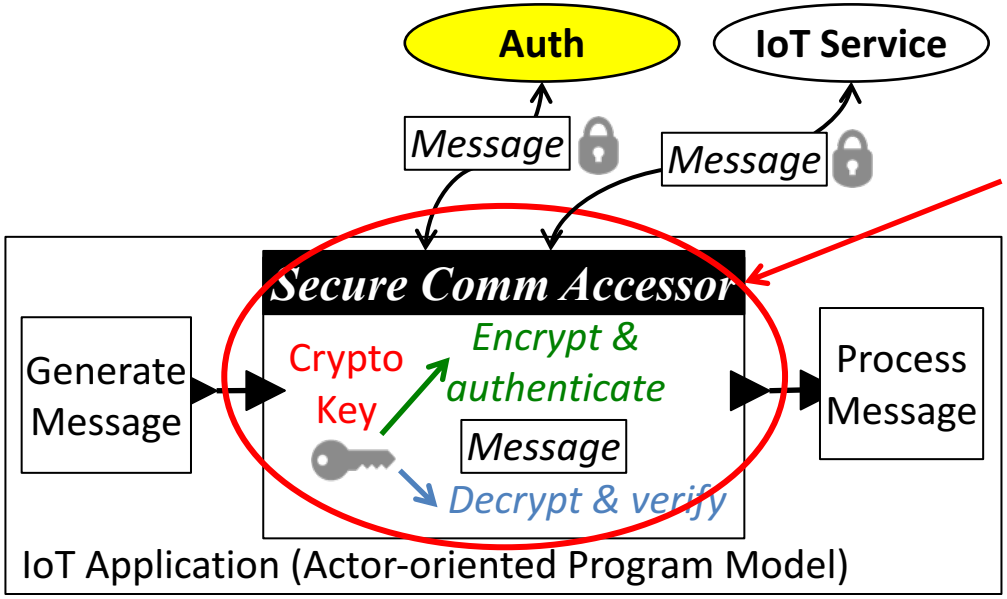


[1] A prototype of Auth has been proposed in Kim et al., 2016. "A Secure Network Architecture for the Internet of Things Based on Local Authorization Entities"

[2] Lopez et al., 2015. "Edge-centric Computing: Vision and Challenges"

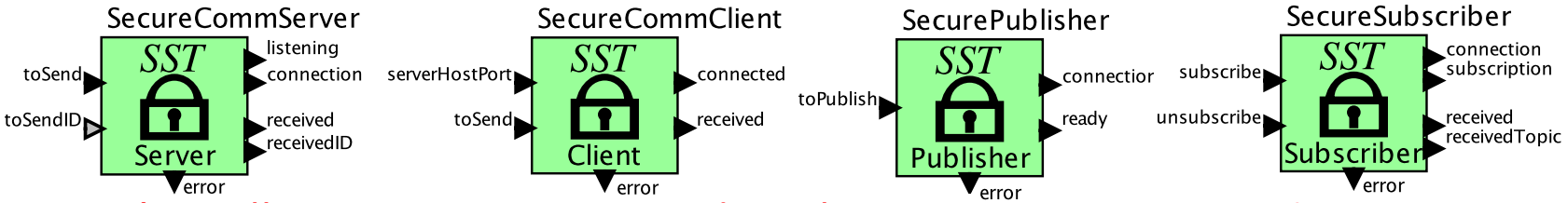
Design and Implementation (cont'd)

- Secure communication accessors



  - Software building blocks for securely accessing Auth and the IoT services
  - Encapsulate crypto keys & operations
  - Help IoT developers who are not security experts

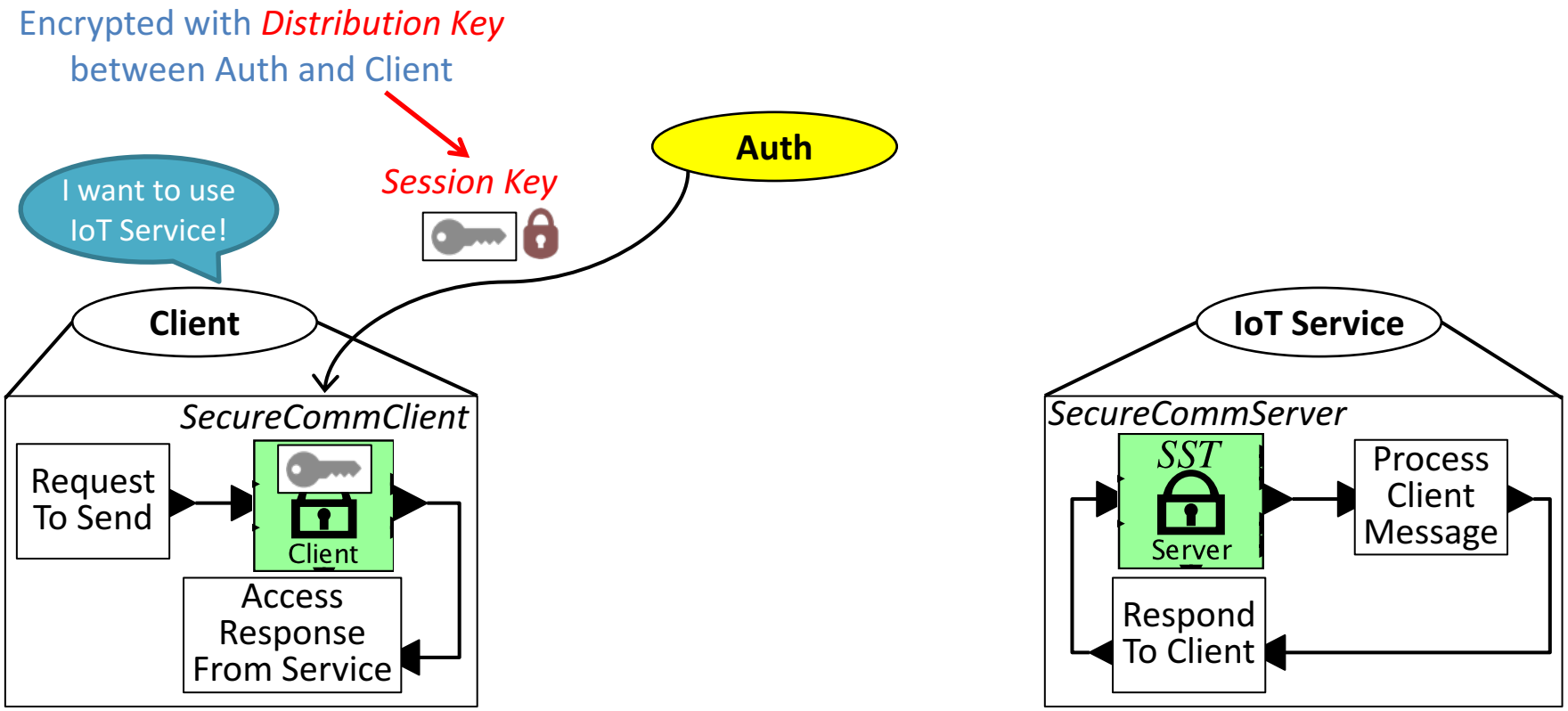
  - Currently available accessors (in JavaScript)



  - We're still at a starting point and working on more accessors!
  - For more information, see https://accessors.org

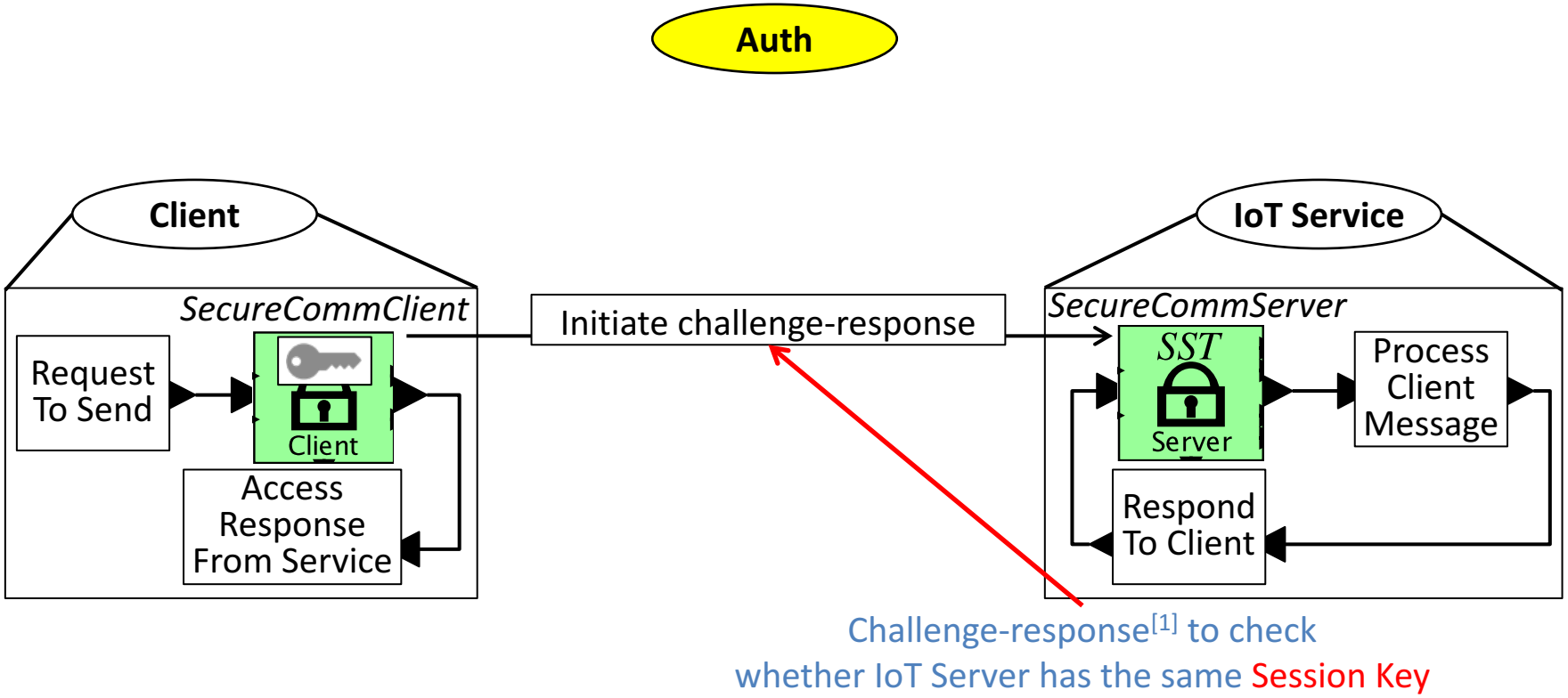
Design and Implementation (cont'd)

- Example: How SST (Auth and accessors) works



Design and Implementation (cont'd)

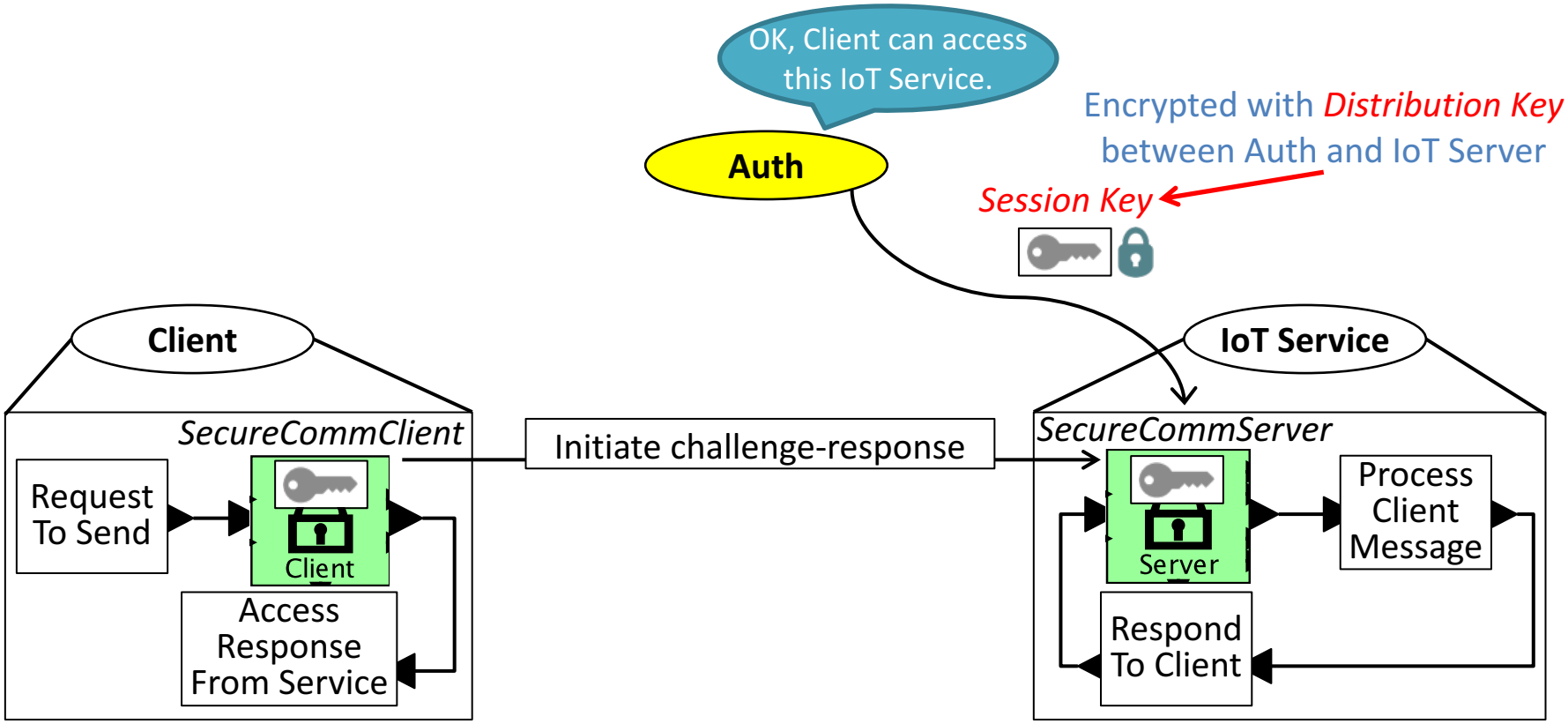
- Example: How SST (Auth and accessors) works



[1] Similar to TLS PSK extension by Eronen and Tschofenig. 2005. Pre-Shared Key Ciphersuites for TLS. RFC 4279.

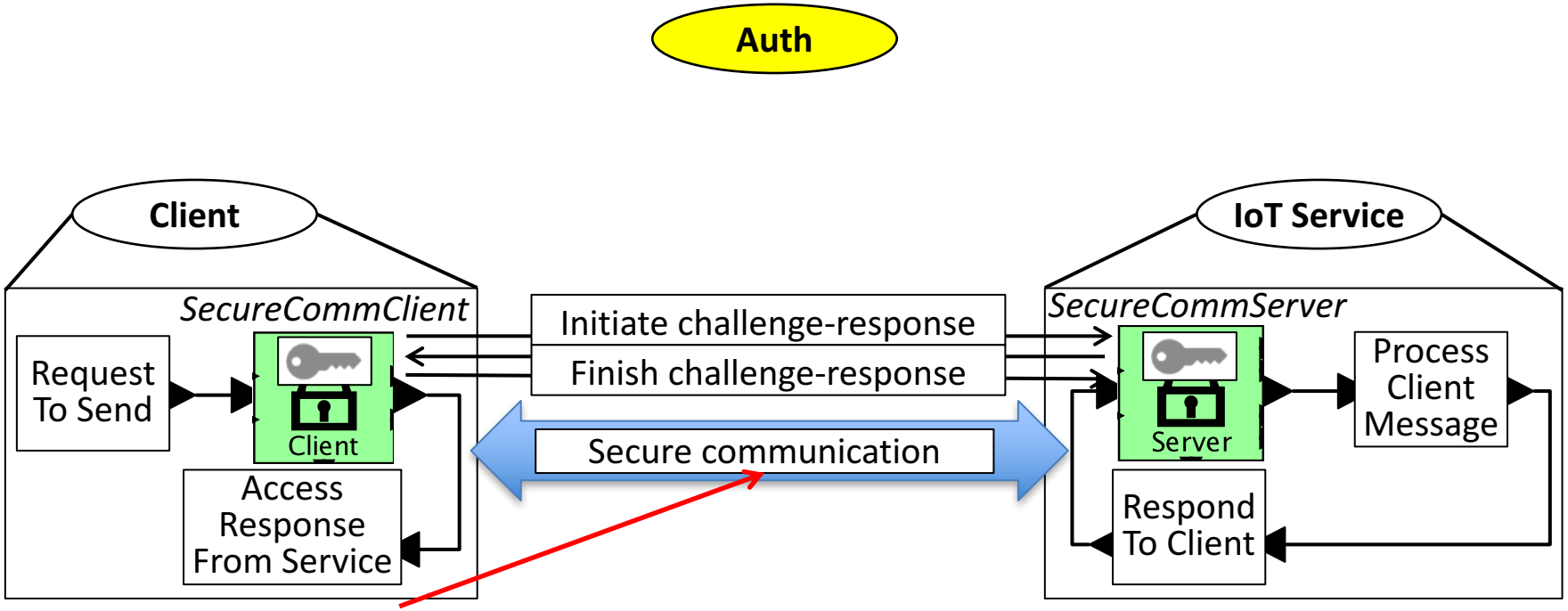
Design and Implementation (cont'd)

- Example: How SST (Auth and accessors) works



Design and Implementation (cont'd)

- Example: How SST (Auth and accessors) works



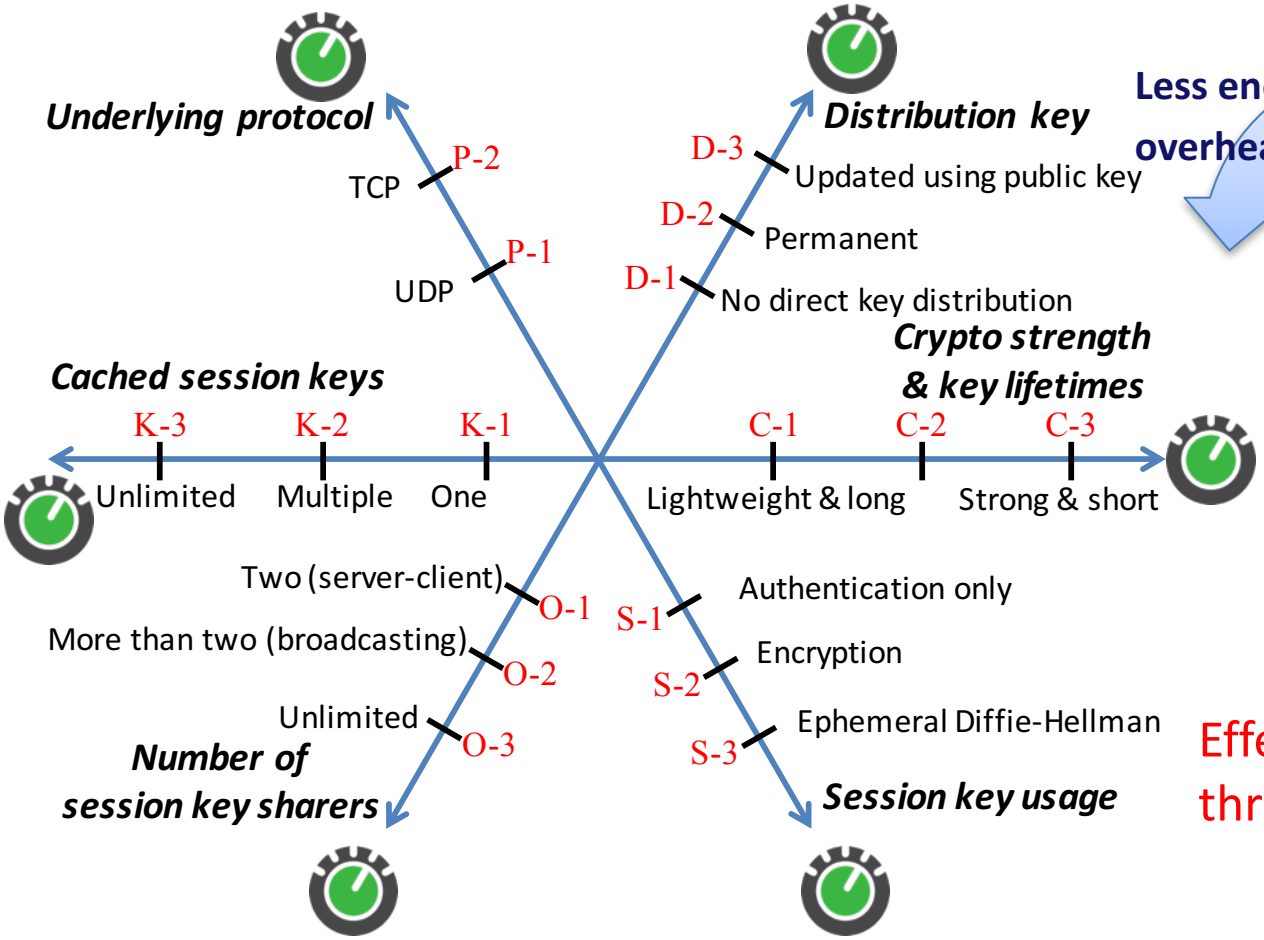
Protected communication channel using session key and standard cryptography[2]

[2] Followed TLS 1.2's standard, including sequence number, encrypt-then-MAC

SST for Heterogeneity

Heterogeneity
Open Env.
Scalability

- SST's configuration alternatives

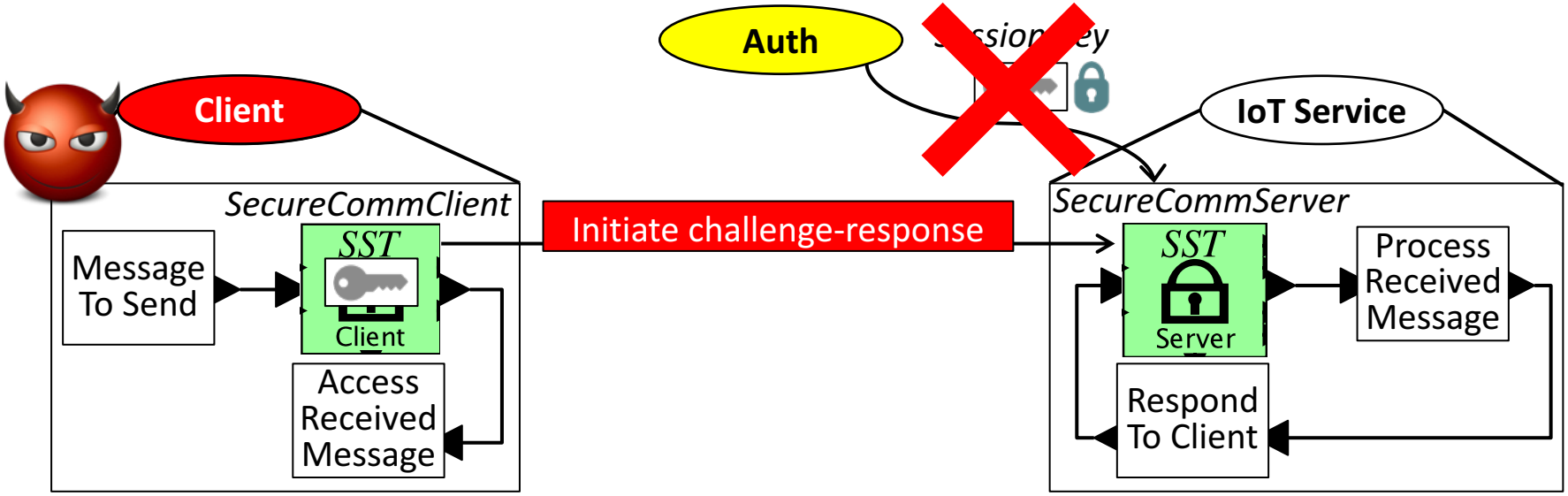


Effect of knobs will be shown through experiments!

SST for Open Environment

Heterogeneity
Open Env.
Scalability

- Limiting damage from compromised entities
  - SST's design to timely revoke keys (session & distribution keys)
    - Must always be authorized by Auth
    - Revocation takes effect immediately

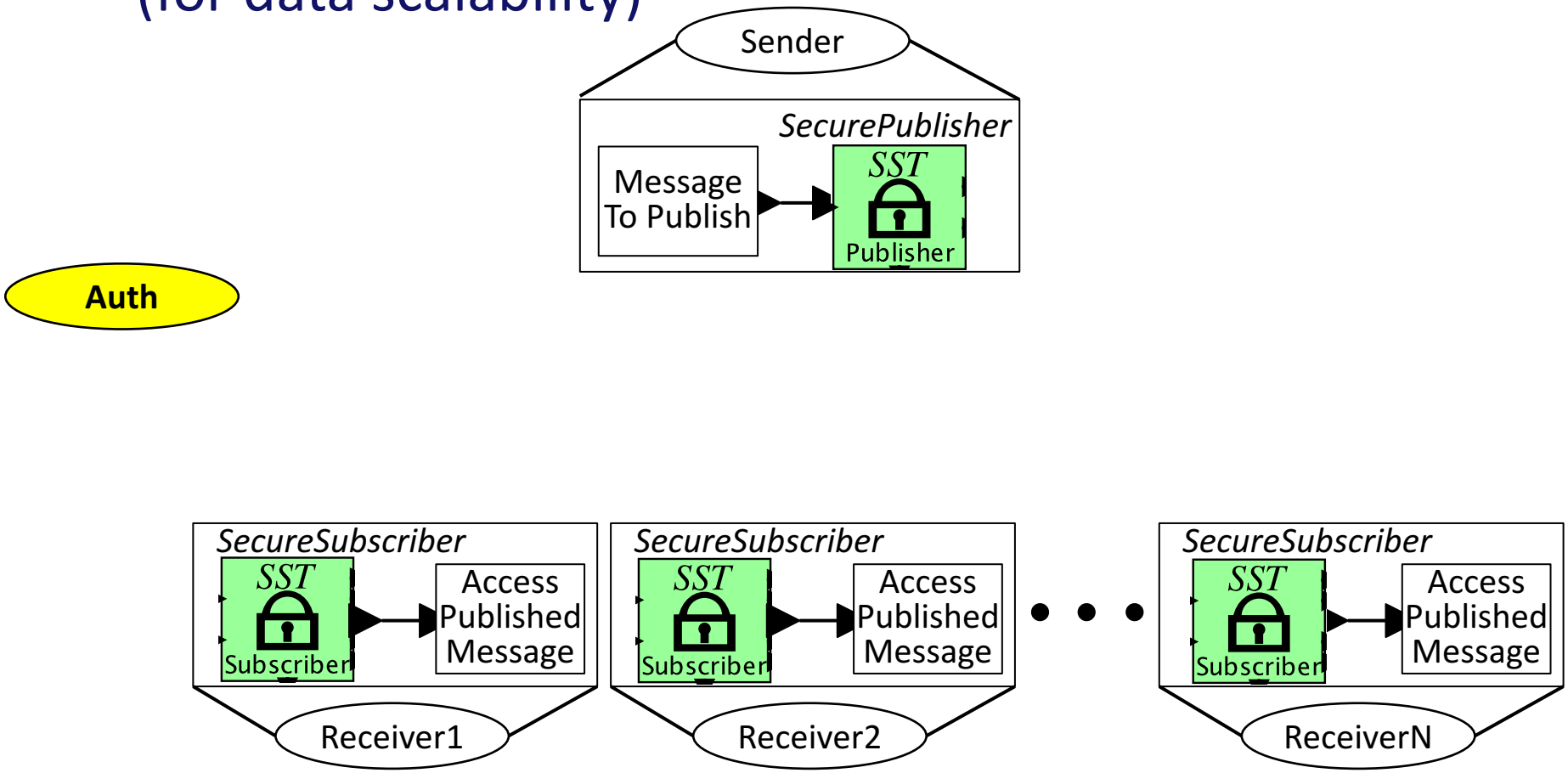


  - Even when Client with a valid session key is compromised, Auth can prevent its access to IoT Server!

SST for Scalability

Heterogeneity
Open Env.
Scalability

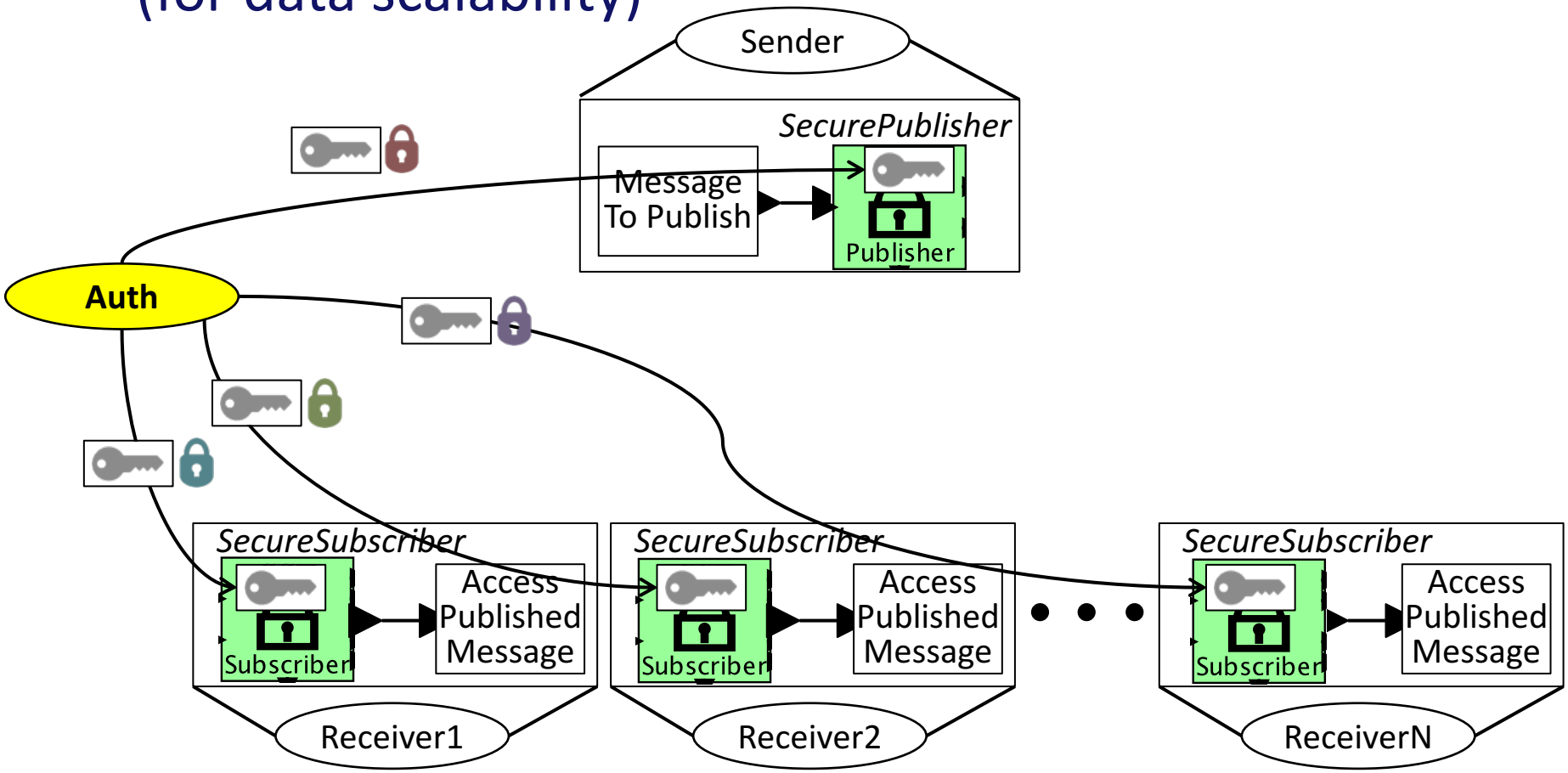
- Shared key support for one-to-many communication (for data scalability)



SST for Scalability

Heterogeneity
Open Env.
Scalability

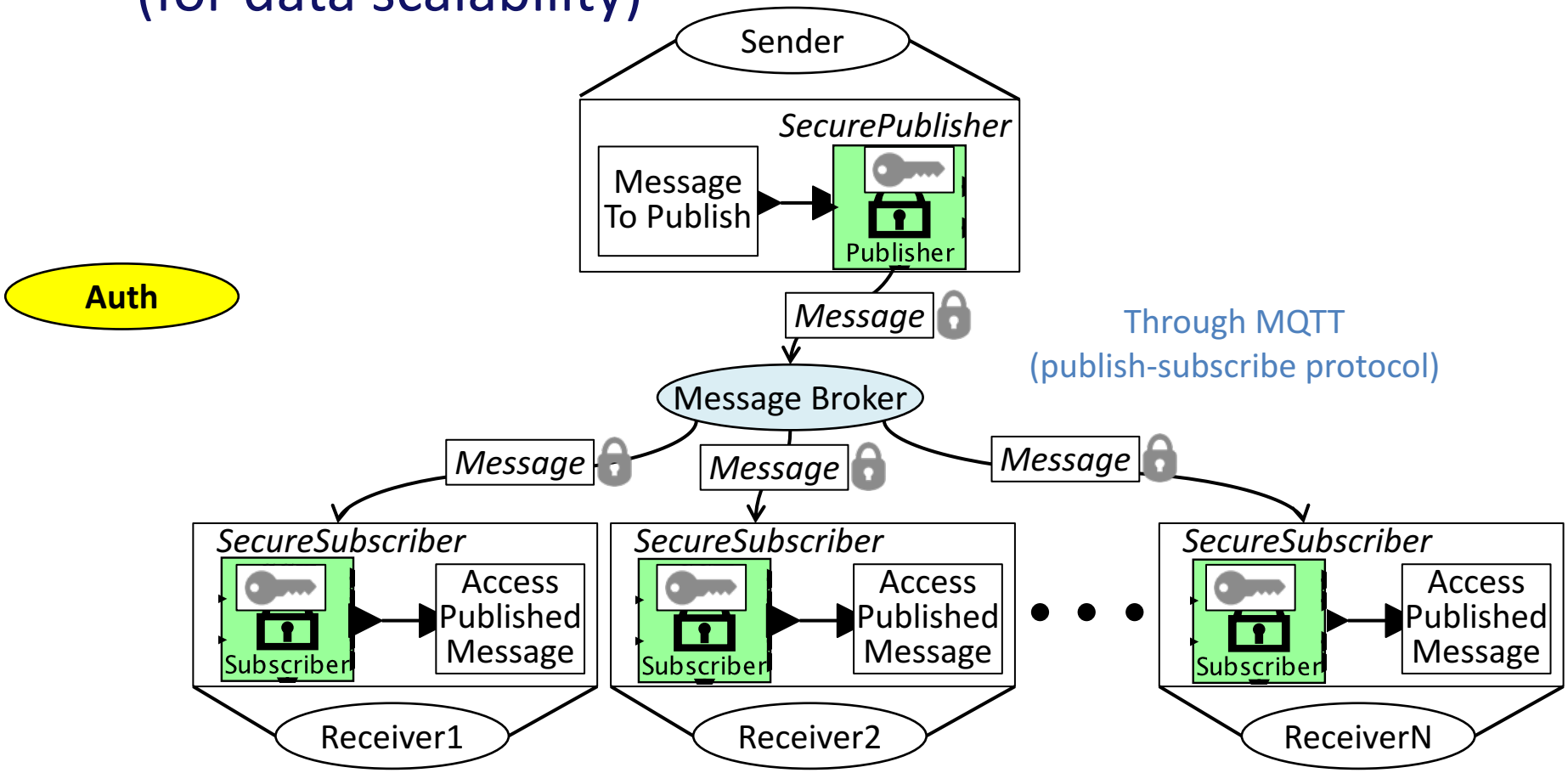
- Shared key support for one-to-many communication (for data scalability)



SST for Scalability

Heterogeneity
Open Env.
Scalability

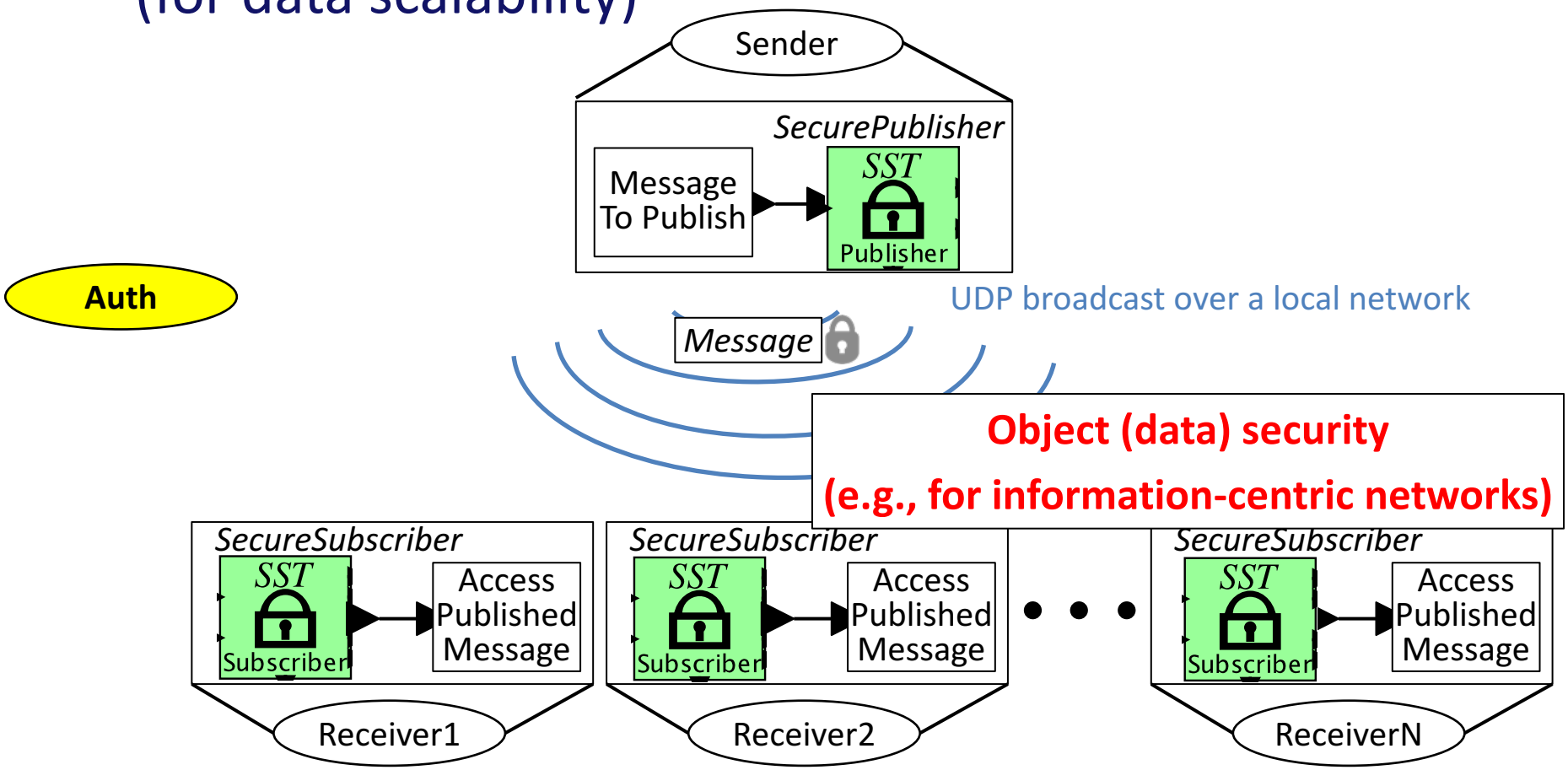
- Shared key support for one-to-many communication (for data scalability)



SST for Scalability

Heterogeneity
Open Env.
Scalability

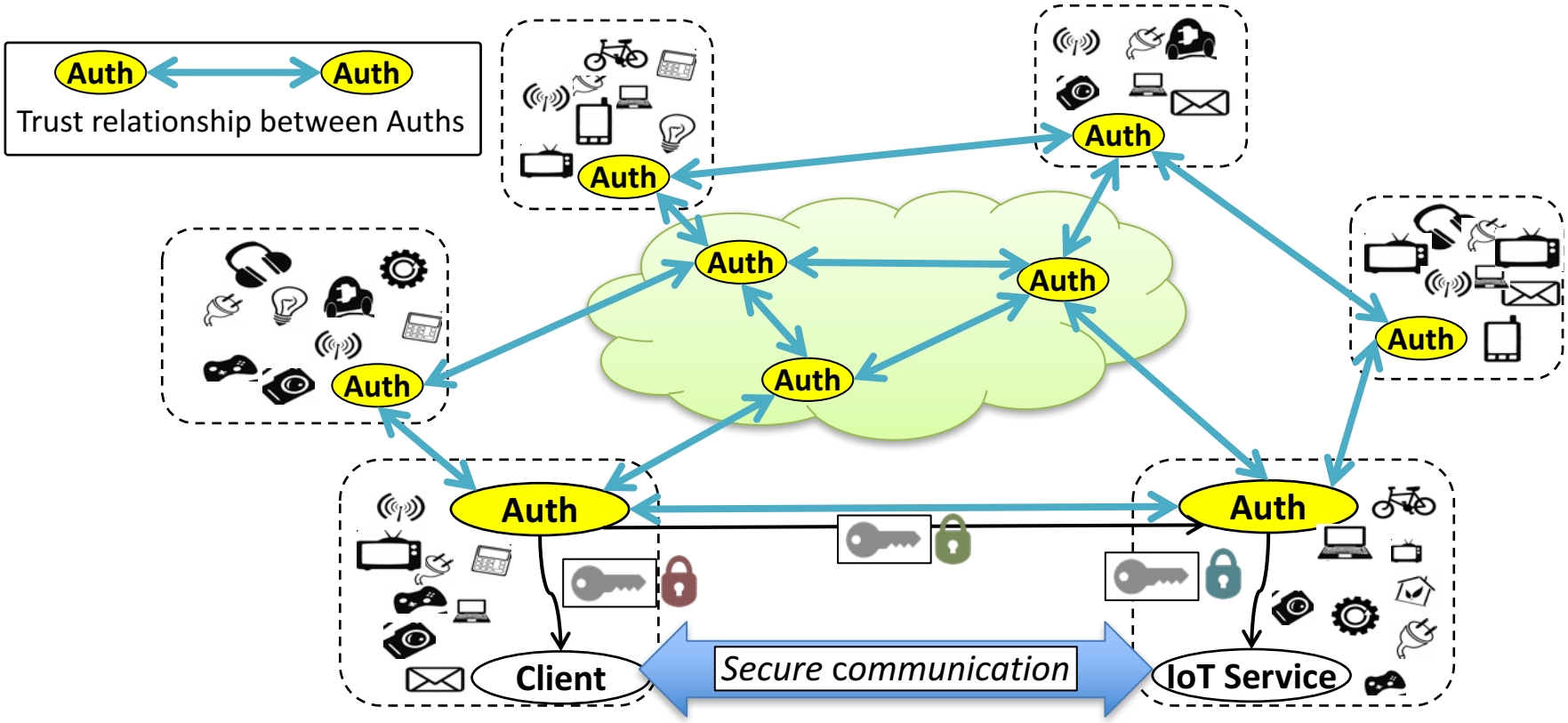
- Shared key support for one-to-many communication (for data scalability)



SST for Scalability (cont'd)

Heterogeneity
Open Env.
Scalability

- Globally distributed Auths (to scale with # IoT devices)
  - Trust relationships without a centralized authority



Evaluation: Security Analysis

Desired Security Properties

- Confidentiality (of data)
- Message authenticity
- Data integrity

Threat Model

- Network attackers
  - Eavesdrop or inject packets
- Compromised IoT Entities
  - Try to break security of others
- No compromised Auths

Formal Security Model of SST[1]

- Modeled in Alloy[2] (Model checking tool & language)
- Includes models for Auths, entities and communication messages

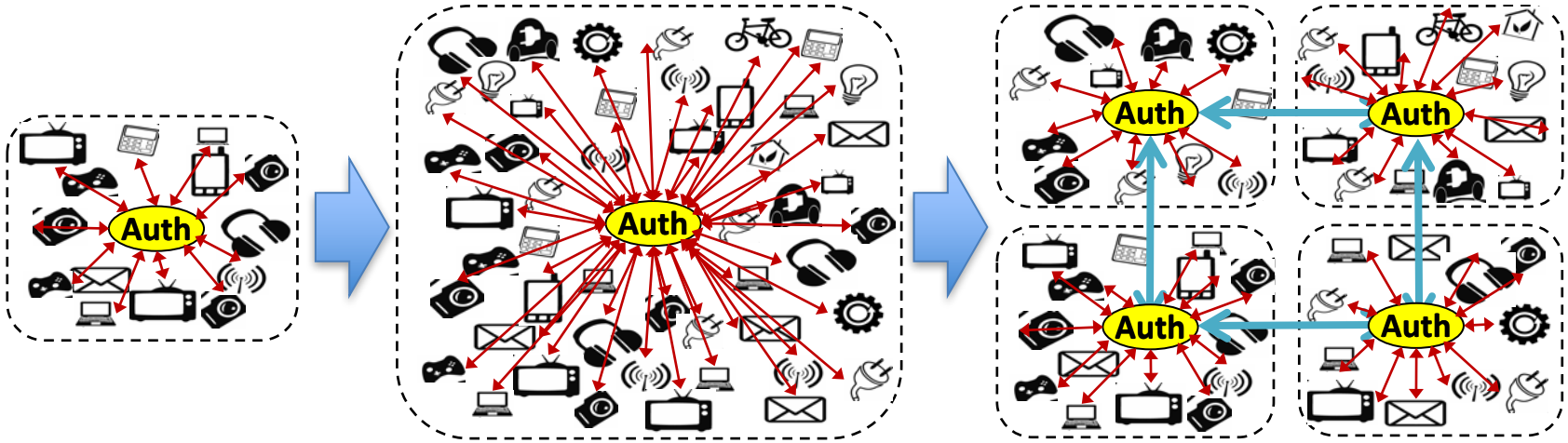
Result: Formally proven to satisfy the security properties!

[1] https://github.com/iotauth/security_analysis
[2] http://alloy.mit.edu/alloy/

Evaluation: Scalability Analysis

- Auth's authorization tasks include
  - Communication with IoT entities and Auths
  - Cryptographic operations
  - Accessing Auth's database (keys, access policy, etc.)

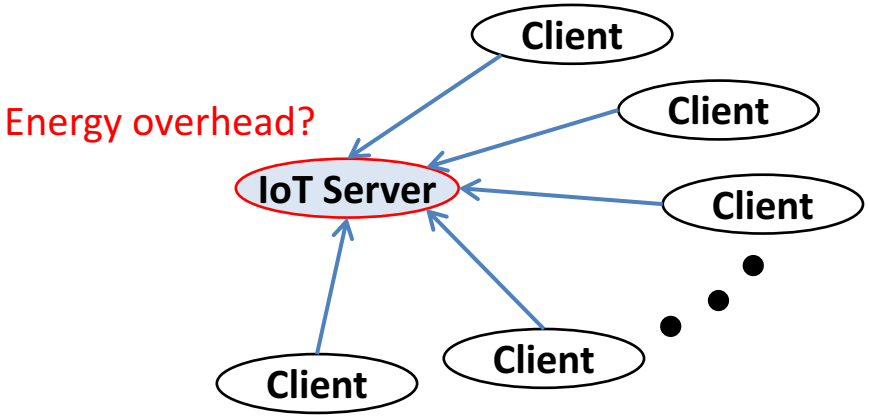
Number of IoT entities ↑
Access activity per entity ↑
Authorization workload ↑



- Scalability analysis result:
  - Each Auth's workload is a linear function of "number of entities per Auth", *not* "total number of entities in the system", assuming access activity per entity is fixed
  - In theory, we can always scale with increasing entities by adding Auths accordingly

Experiments & Results

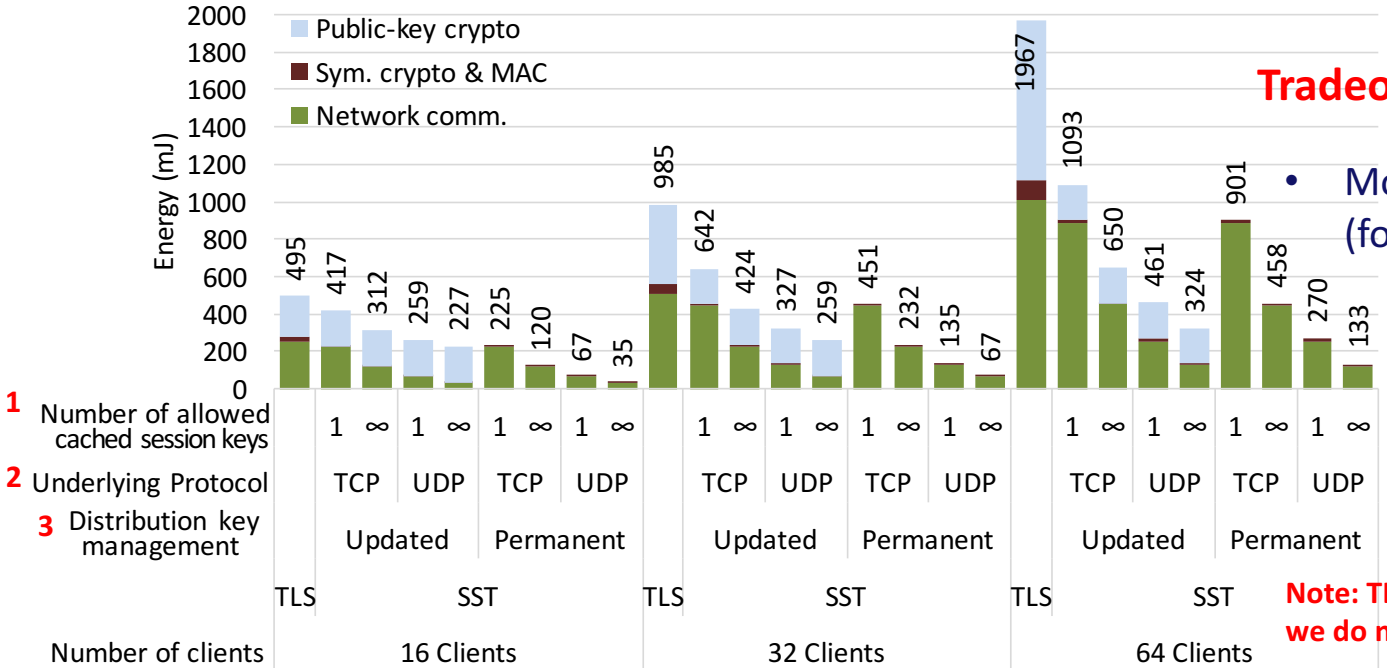
- Effect of various configuration alternatives
  - Estimated energy consumption for setting up secure connections between IoT clients & IoT servers
    - Logged crypto operations and captured packets
    - Used energy numbers from UAB[1] and SICS[2]



[1] UAB (Universitat Autònoma de Barcelona), Rifà-Pous and Herrera-Joancomartî. 2011
[2] SICS (Swedish Institute of Computer Science), Feeney and Nilsson. 2001

Experiments & Results (cont'd)

Estimated energy for an IoT server connected by 16, 32, and 64 clients

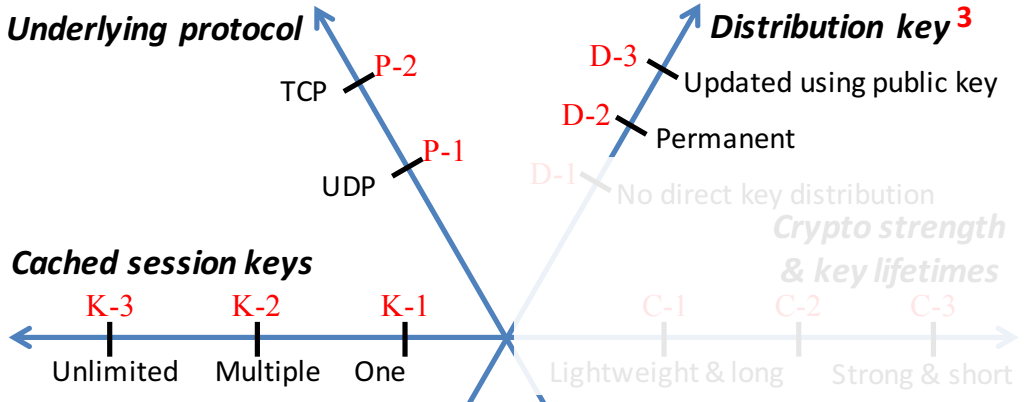


Tradeoffs for heterogeneity!

- More results in our paper! (for IoT clients)

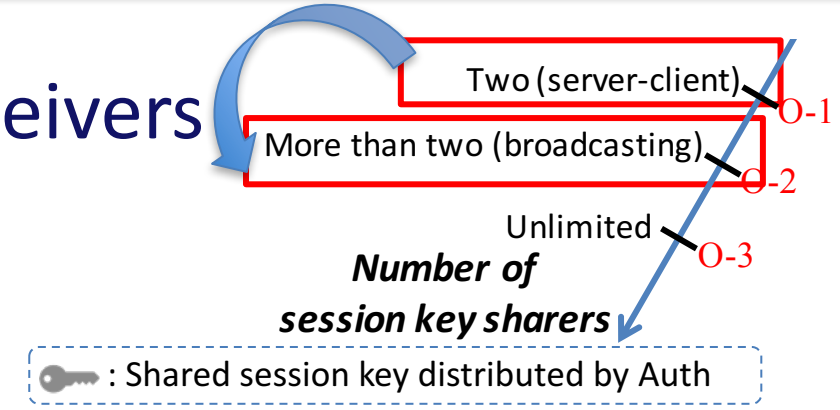
Note: TLS was used as a reference and we do not claim that SST is better than TLS

Less energy overhead — **More security guarantees**

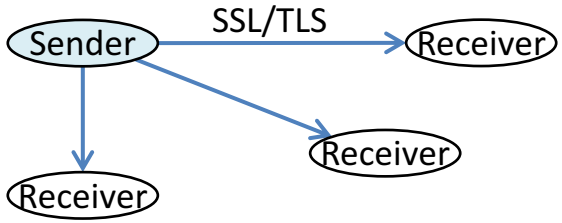


Experiments & Results (cont'd)

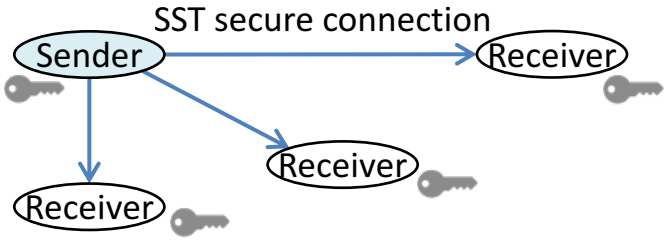
- A sender and multiple receivers with different settings



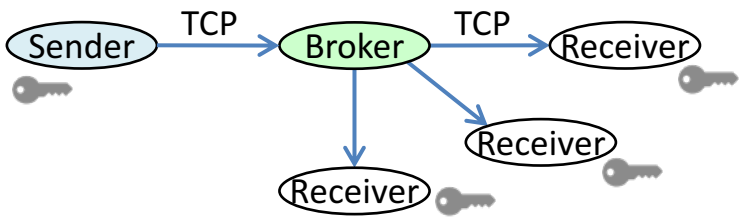
(1) Connections with SSL/TLS



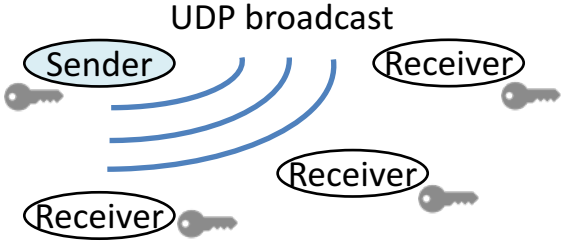
(2) Shared Key + secure connections by SST



(3) Shared key + MQTT message broker

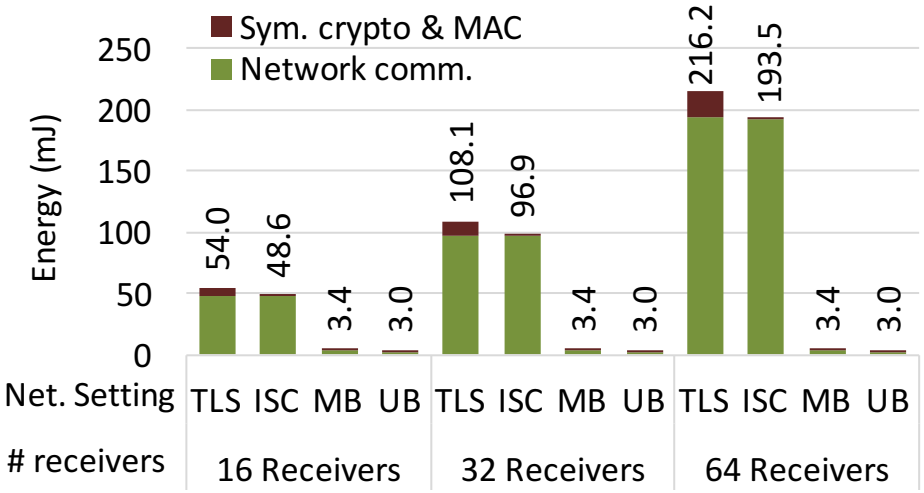


(4) Shared key + UDP broadcast



Experiments & Results (cont'd)

Estimated energy for a sender to send a 1KB-message to receivers



TLS: (1) SSL/TLS

ISC: (2) Individual SST Connections + shared key

MB: (3) MQTT Message broker

UB: (4) UDP broadcast

- More results in our paper! (for sender initialization)

Tradeoff example

A sensor node (500mAh/1.5V battery) sending 1KB per minute to 64 receivers

Expected battery life

<10 days with ISC (secure connections by SST)

625 days with UB (UDP broadcast)

Image: DevDuino Sensor Node V1.3

Conclusions

- Benefits of SST: Secure Swarm Toolkit
  - Authorization for a wide range of IoT from sensor nodes to safety-critical systems
  - Enable Internet-scale deployment with increasing connected devices and traffic
  - Help deployment of IoT security solutions by system designers with moderate knowledge in security
  - Possible integration with other IoT-related efforts (e.g., securing CoAP[1])

[1] Shelby et al., 2014. "The Constrained Application Protocol (CoAP)". RFC 6347

Conclusions (cont'd)

- Future work
  - Mitigation against availability attacks (e.g., Denial-of-Service attacks)
  - Detection of malicious behavior of compromised IoT entities or Auth
  - Further studies on usability of SST
  - Efficient initial setup of SST (e.g., registering IoT devices with Auth)
- For further information
  - https://github.com/iotauth