A Secure Network Architecture for the Internet of Things Based on Local Authorization Entities

Hokeun Kim, Armin Wasicek, Benjamin Mehne and Edward A. Lee
Dept. of EECS, University of California, Berkeley

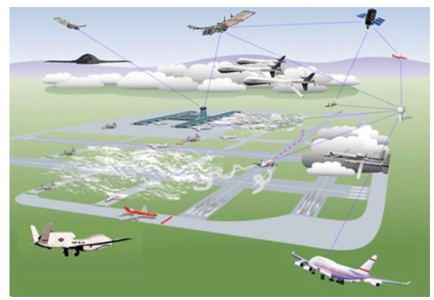
FiCloud 2016, August 22-24, 2016, Vienna, Austria

Challenges for IoT Security

- **Heterogeneity** in security requirements & resource availability
  - Examples of heterogeneity in the IoT

Cardiac monitor and emergency service

- Privacy
- Resource constraints consideration



Drones and ground air traffic control

- Strong and frequent authorization (safety-critical)
- Intermittent connectivity consideration

Apple pay

- Confidentiality
- Authentication
- Moderate resource constraints consideration

Ambient temperature sensors and receiver

- Data integrity
- Resource constraints consideration

- Images from www.dicardiology.com, diydrones.com, en.wikipedia.org, and safesoundfamily.com
- IoT security challenges from Miorandi *et al.*, AD Hoc Networks, 2012, Jing *et al.*, Wireless Networks, 2014, Sadeghi *et al.*, DAC, 2015

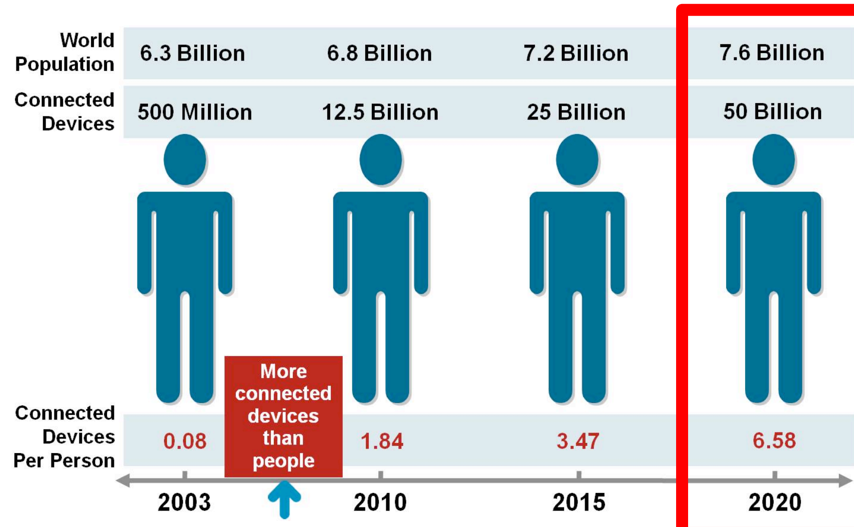
Challenges for IoT Security

- Operation under **open/untrusted environment**
  - More remotely/physically accessible

Traffic lights and controller in Ann Arbor, Michigan

- Ghena *et al.*, “Green Lights Forever: Analyzing the Security of Traffic Infrastructure,” WOOT 2014.

- **Scalability**
  - Security solutions for IoT should be scalable!



- Source: Cisco IBSG, April 2011

- IoT security challenges from Miorandi *et al.*, AD Hoc Networks, 2012, Jing *et al.*, Wireless Networks, 2014, Sadeghi *et al.*, DAC, 2015

IoT-related Security Requirements Breakdown

- Frequent authorization/authentication
- Automated mutual authentication
- Dealing with intermittent connectivity
- Support for one-to-many communication (for scalability)
- Consideration for resource constraints
- Privacy
- Dynamic entity registration

Goals & Contributions of Proposed Network Architecture

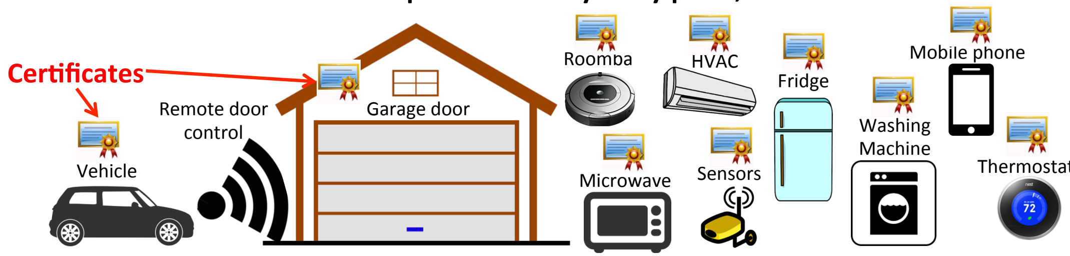
- Network-level approach using encryption/secure hash and authorization control over the Internet of Things

- New cipher/hash algorithms, new authentication/authorization system, new key management techniques? **No!**

- **Organization and integration** of existing approaches with emphasis on flexibility and usability
  - Specifically, to address IoT-related security requirements in the previous slide

Issues with Applying Widely-Used SSL/TLS for All Devices in the IoT

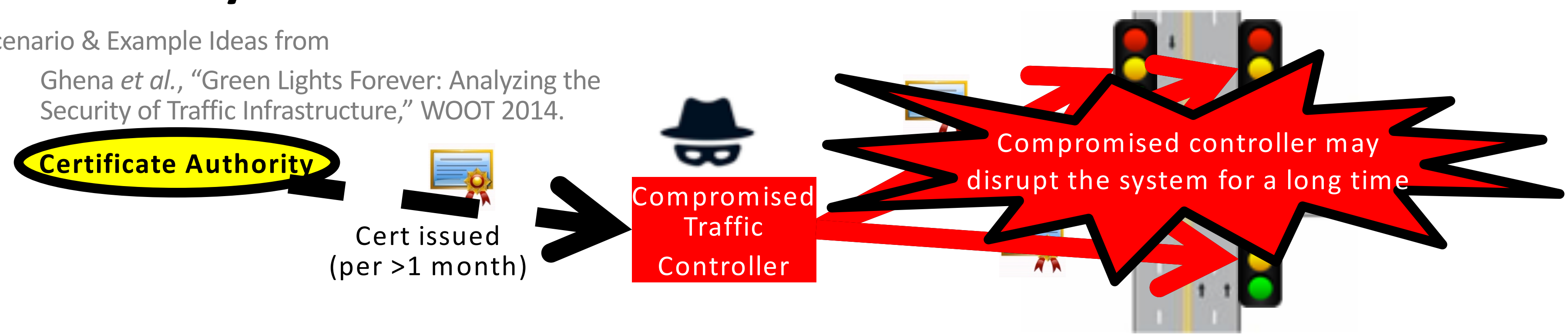
- Overhead for **resource-constrained devices**
  - Energy/computation overhead for public key crypto, communication bandwidth, memory, etc.



- Limited support **one-to-many communication** (e.g., pub/sub)
  - Connections are based on 1-to-1 connections (server/client model)
- **Security issue** for entities **under untrusted environment**

Scenario & Example Ideas from

- Ghena *et al.*, "Green Lights Forever: Analyzing the Security of Traffic Infrastructure," WOOT 2014.



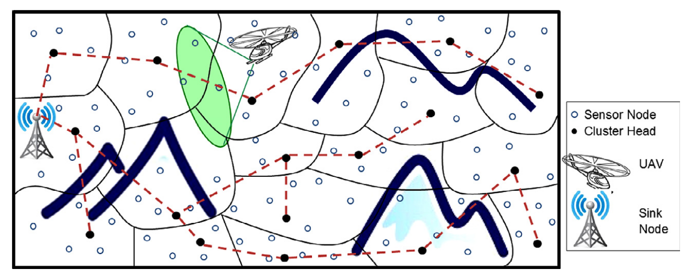
  - Once a certificate is issued, CA does not control connections between entities

Kerberos and Approaches in WSNs

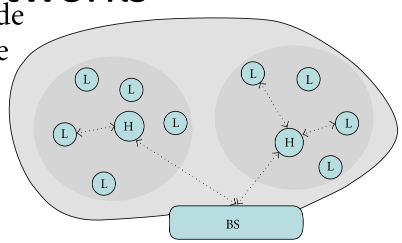
- **Kerberos authentication system**[1] provides direct control over connections between entities by issuing temporary tickets for authentication
- However, Kerberos has limited support for automated authentication and intermittent connectivity

[1] C.Neuman, T.Yu, S.Hartman, and K.Raeburn, “The Kerberos Network Authentication Service (V5),” RFC 4120, IETF, Jul. 2005.

- Huang *et al.*, 2011
  - For hierarchical/heterogeneous sensor networks
- Sahingoz, 2013
  - Using UAV for authentication of large-scale WSNs



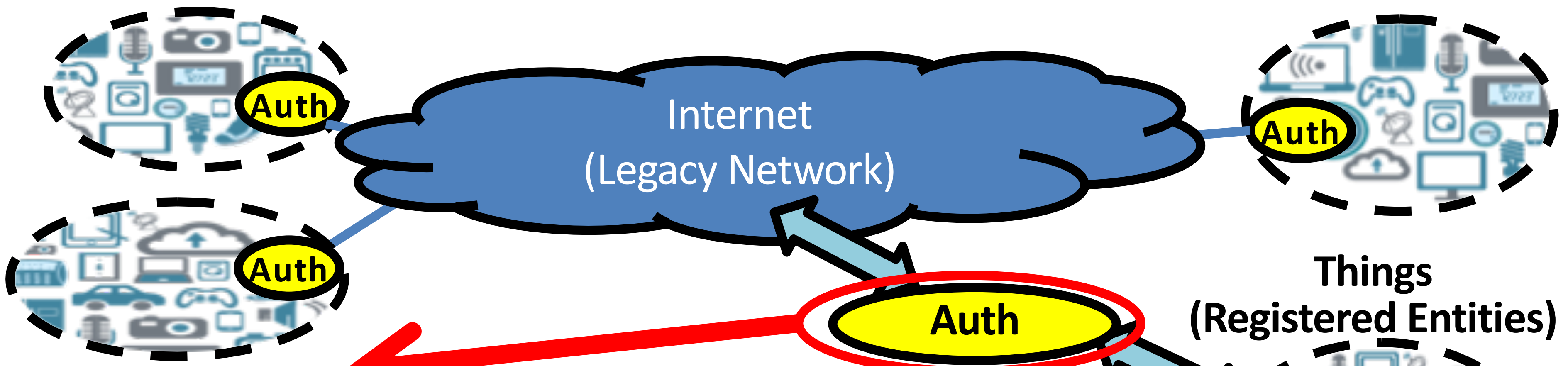
BS: Base station
H: H-sensor node
L: L-sensor node



- Erfani *et al.*, 2015
  - A key management system for dynamic addition/deletion of mobile nodes

Proposed Architecture Overview

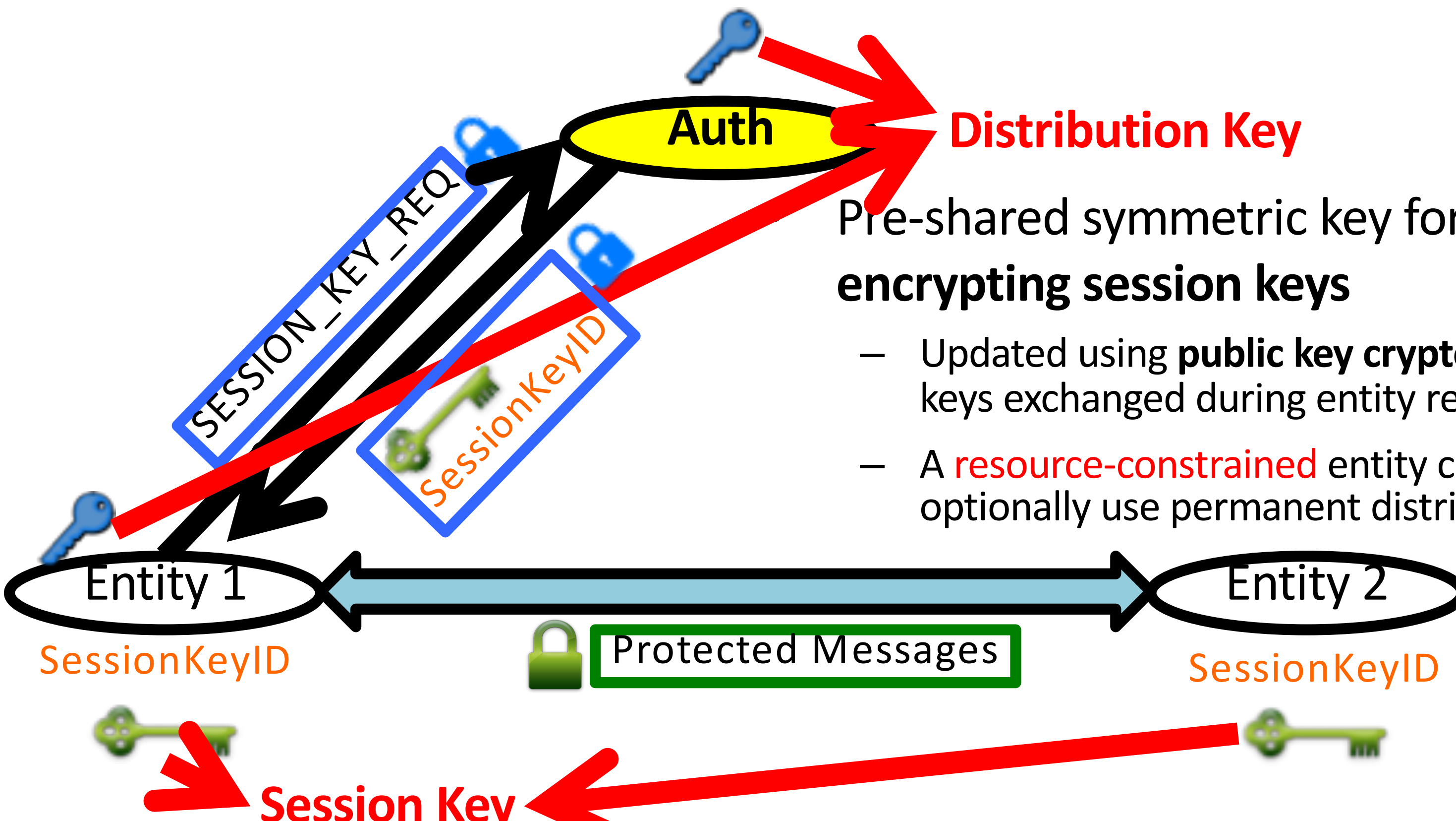
Auth – Local Authorization Entity



- **Local point of auth**(authentication/authorization) & trusted third party for locally registered entities
- Make communication authorization locally manageable
- Fully automated except for entity registration
- Deployed on bridges between the **Internet** and **Things** (e.g., Intel IoT gateways)

Proposed Architecture Overview

Keys Used in Proposed Architecture



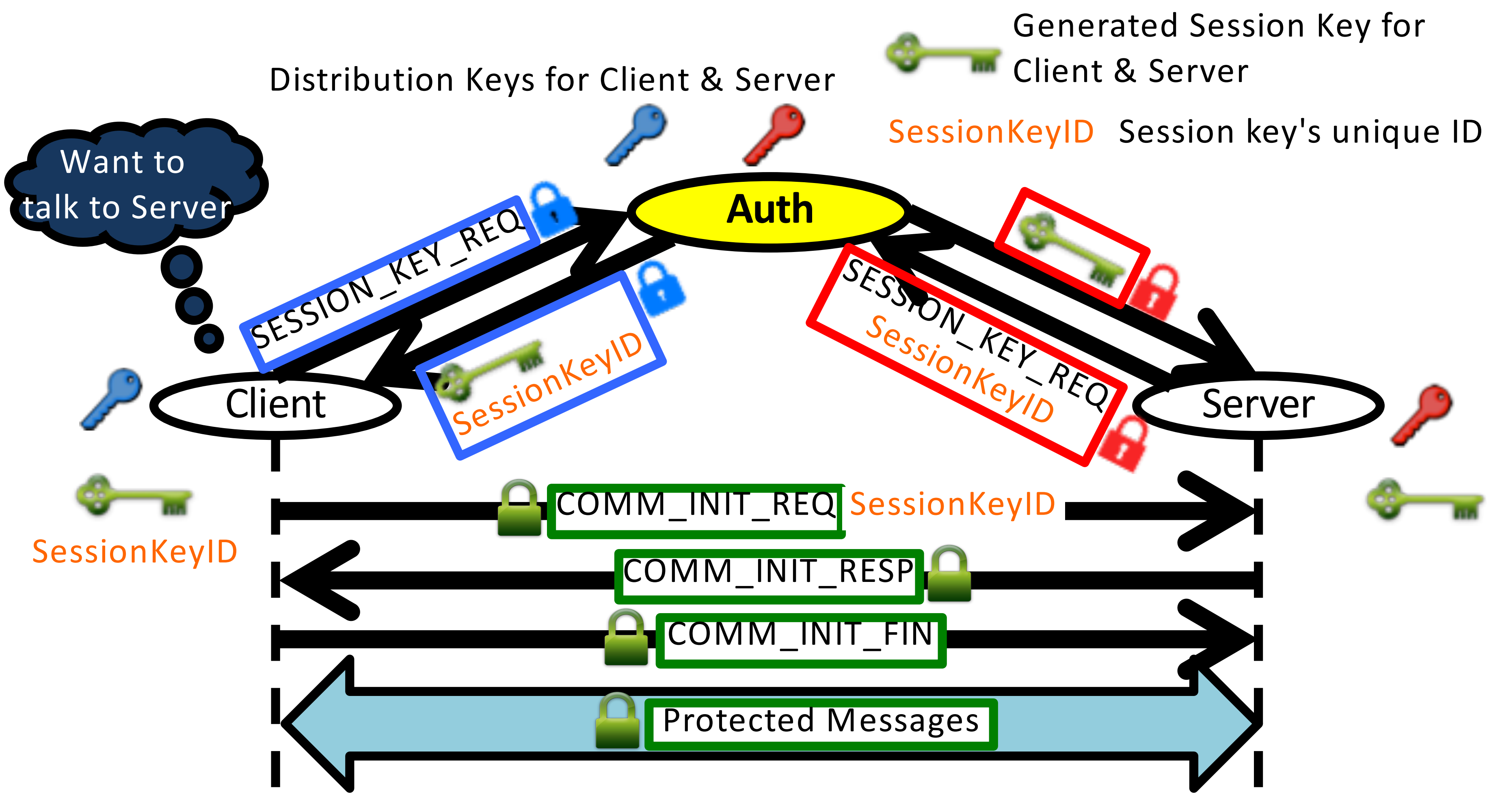
Pre-shared symmetric key for **encrypting session keys**

- Updated using **public key crypto** (public keys exchanged during entity registration)
- A resource-constrained entity can optionally use permanent distribution

- Symmetric crypto key for **protecting a single session** of communication
  - Given to only authorized devices
  - Unique **Session key ID**, including ID of Auth who generated it

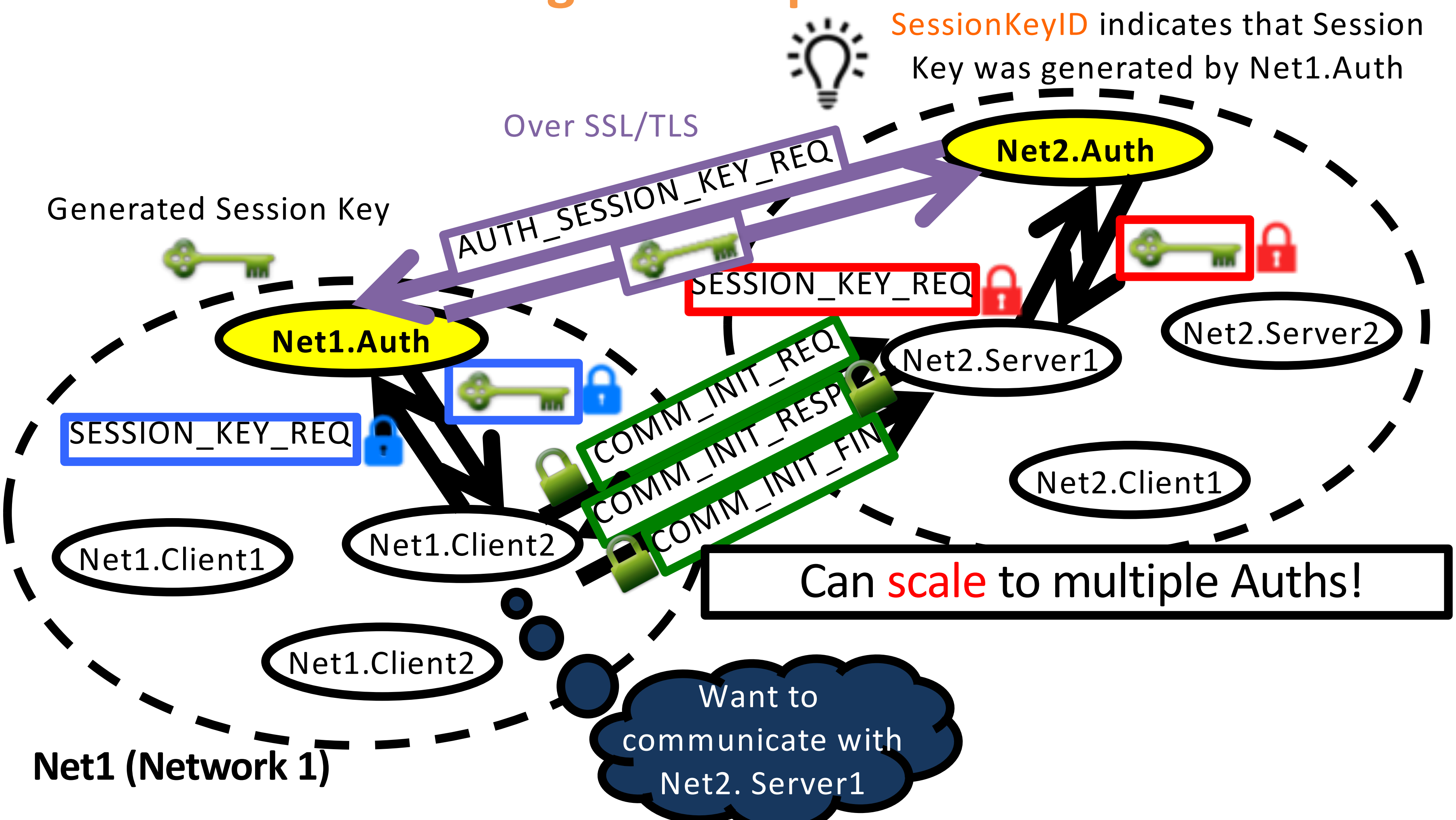
Proposed Architecture Overview

Operation Phases – With an Example Scenario



Proposed Architecture Overview

Scaling To Multiple Auths



Proposed Architecture Overview

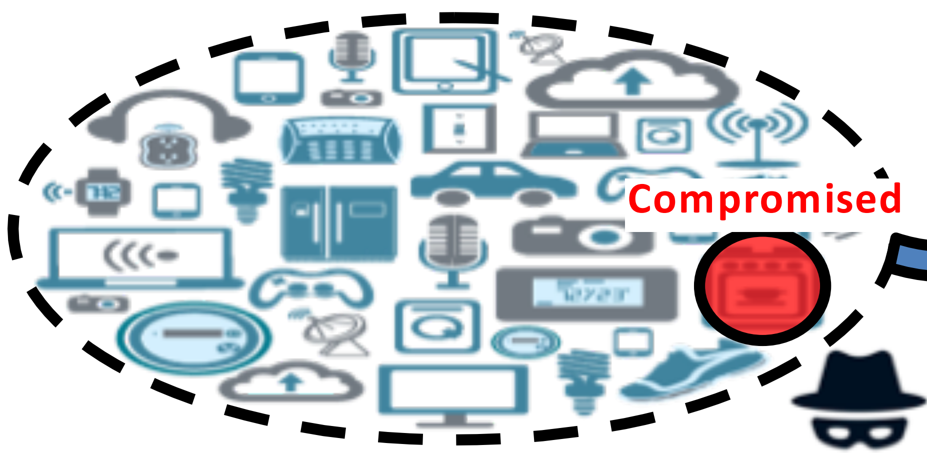
Auth DB – Information for Local Authorization

Main Database Tables

- Registered Entity Table
- Communication Policy Table
- Cached Session Key Table
- Trusted Auth Table

Auth DB

Auth



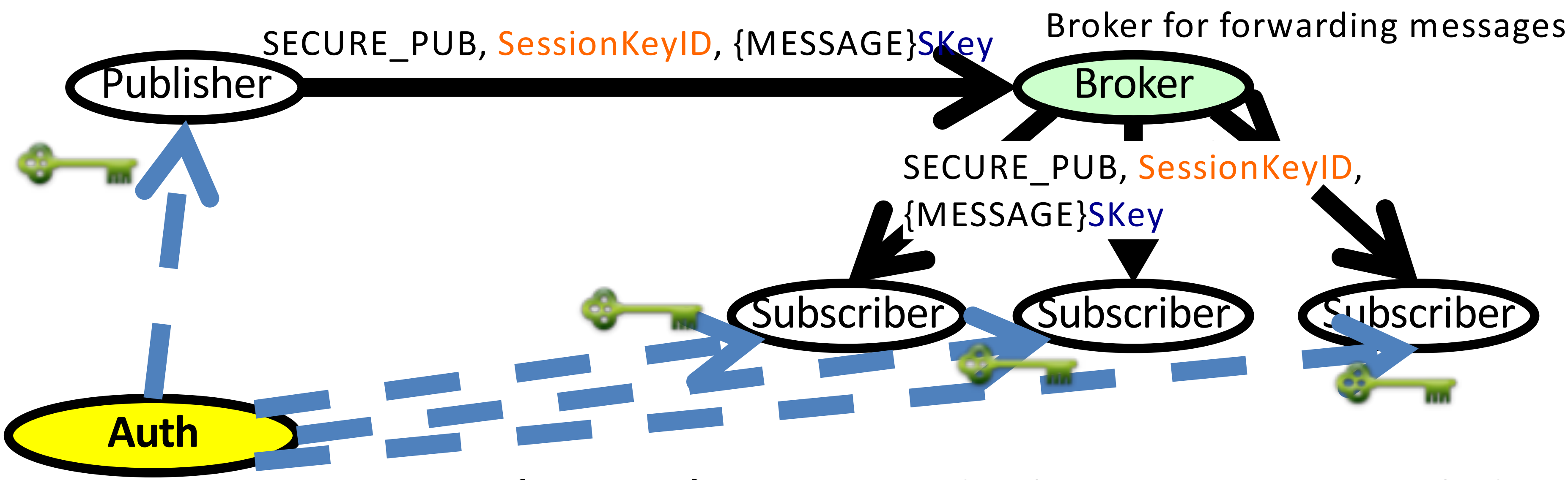
Auth DB will be updated to revoke credentials

- Invalidate cached keys
- No more keys for authorization

If an intrusion or a compromised entity is detected

Support for One-to-Many Communication

- Distributing shared key for securing one-to-many communication (e.g. broadcasting, publish-subscribe protocol such as MQTT)



- **{MESSAGE}SKey:** Encrypted with Session Key, MAC attached

- Possible **integration** with one-to-many communication authentication protocols, such as TESLA[1] (Timed Efficient Stream Loss-tolerant Authentication)

[1] A. Perrig, R. Canetti, J. Tygar, and D. Song, "The TESLA Broadcast Authentication Protocol," RSA CryptoBytes, 2005.

Experimental Setup & Implementation

- Compare security overhead with **SSL/TLS**
  - Widely used, can support strong crypto including public key crypto
  - Compare security overhead (crypto operations & sent/received packets)
- Prototype implementation
  - Use **Node.js** to implement Auth and Entities for Proposed and SSL/TLS
  - Modify **OpenSSL library** included in Node.js to capture crypto operations
- Overhead ➔ Energy consumption (energy numbers are obtained from [1],[2])

Same cipher/hash algorithms for **Proposed** and **SSL/TLS**

Operation	Energy cost
RSA-2048	91.02 mJ per encrypt/sign operation
	4.41 mJ per decrypt/verify operation
AES-128-CBC	0.19 µJ per byte encrypted/decrypted
SHA-256	0.14 µJ per byte digested
Send* packet	454 µJ + 1.9 µJ × packet size (bytes)

[1] Rifa-Pous and Herrera-Joancomarti, "Computational and Energy Costs of Cryptographic Algorithms on Handheld Devices," Future Internet, 2011.

[2] Feeney and Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," INFOCOM 2001

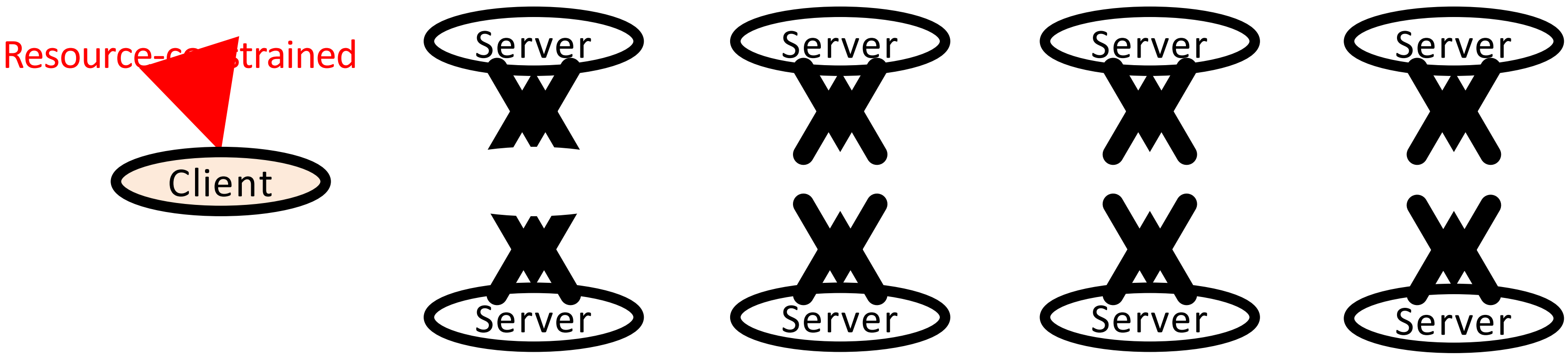
❖ Energy consumption for crypto operations measured on a PDA, HP Hx2790

❖ Energy consumption for IEEE 802.11 … WaveLan … onment

Experiments are carried out using two scenarios that can occur frequently in the IoT

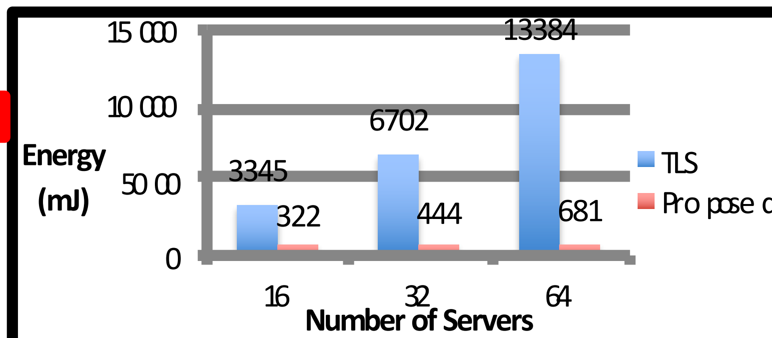
Scenario 1 Experiments & Results

- **Scenario 1**
  - A **resource-constrained client** establishes secure connection with **servers**



- **Results**: client setup/close with 16, 32, 64 servers

Number of Servers	16 servers		32 servers		64 servers	
Approaches	TLS	Proposed	TLS	Proposed	TLS	Proposed
RSA-2048 (Enc/Decrypted)	32/32	2/2	64/64	2/2	128/128	2/2
AES-128-CBC (Bytes)	5,120	3,744	10,240	7,392	20,480	14,688
SHA-256 (Bytes)	188,976	1,957	377,952	3,349	755,904	6,133
Packets (Sent/Received)	159/145	135/120	332/300	263/232	650/587	511/449
Sent Bytes	56,168	11,031	113,120	21,143	222,502	40,735
Received Bytes	66,808	9,453	134,176	17,805	263,956	34,023

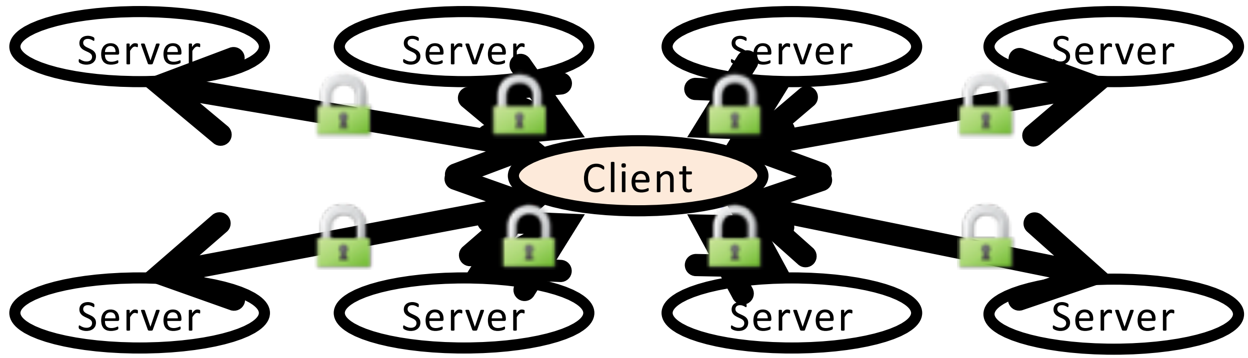


Less energy, by optimizing the use of crypto algorithms

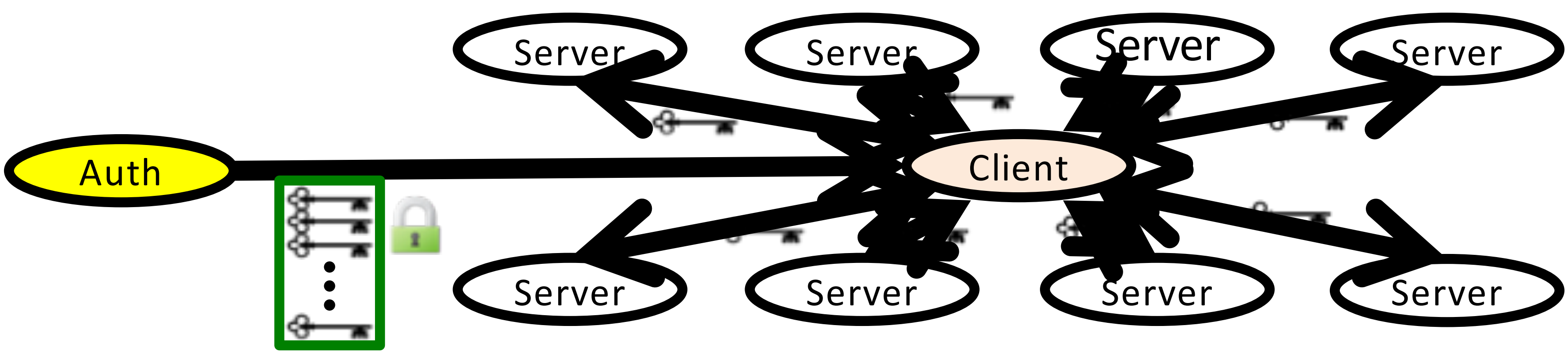
Scenario 1 Experiments & Results

: Public key crypto operations

- **SSL/TLS** – each connection with servers needs public key operations



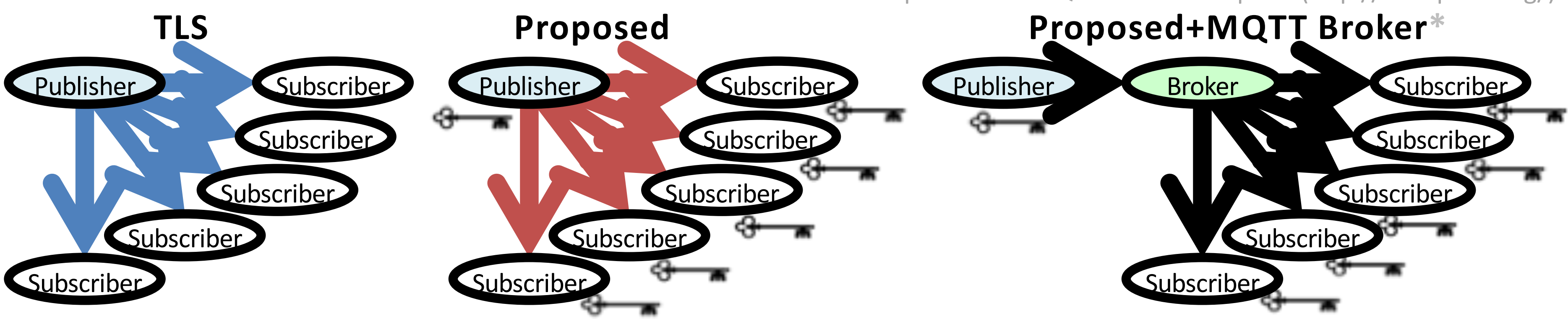
- **Proposed** – public key operations only necessary for communication with Auth



Scenario 2 Experiments & Results

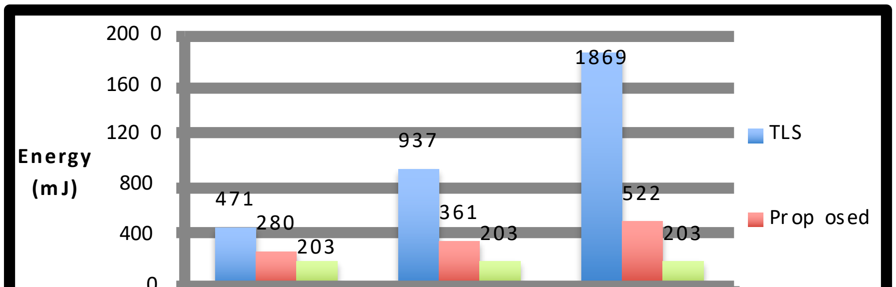
- **Scenario 2**
  - A **resource-constrained publisher** publishes a message that **only authorized subscribers** can read

* Open source MQTT broker Mosquitto (http://mosquitto.org/)

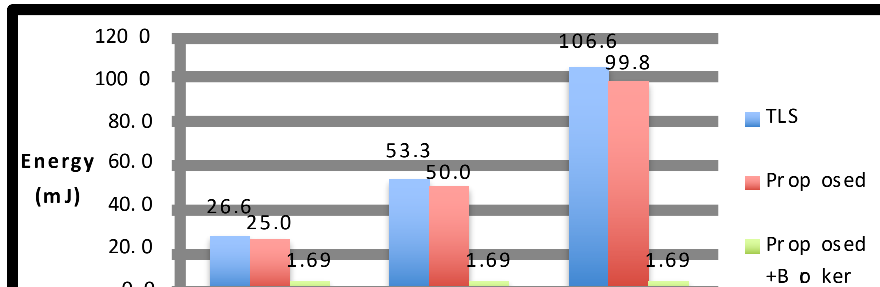


: Shared session key for a pub-sub topic

- Estimated energy for **publisher setup**



- For **publishing** a 256-byte message



Saves not just crypto operations, but also communication

Wrap-up Discussion

- How proposed approach can meet IoT security requirements

Requirements	Proposed Approach
Frequent authentication/authorization	– Auth controls every communication, short key validity period
Automated mutual authentication	– No human intervention required
Intermittent connectivity	– Use of cached keys
Support for scalability features	– Shared session key for more than two entities for publish-subscribe
Consideration for resource constraints	– Use of small, lightweight symmetric session key for authentication
Privacy	– No unique identifier for authentication (temporary session key)
Dynamic entity registration	– Register/unregister can be done within Auth

Conclusion & Future Work

- Proposed **secure network architecture** based on local authorization entities, which can
  - Integrate existing network security measures
  - Address IoT-related heterogeneous security requirements
- Implemented **prototypes** of proposed approaches, obtained preliminary but **promising results**
- Currently working on
  - An open-source implementation of Auth in Java, and example codes for entities in various programming languages
  - Security analysis of protocol
  - Building software components for accessing Auth service

Thank you!

- Q & A

- Contact
  - hokeunkim@eecs.berkeley.edu
  - https://eecs.berkeley.edu/~hokeunkim

- Open-source project page
  - https://github.com/iotauth