Securing IoT Networking with ICN

Dhruti Lalaji Sawant (Matrikel Nummer: 5156829)

***Abstract*—Internet of Things (IoT) is connecting our physical world with the internet e.g. connected automibiles, e-Health, smart homes , industrial automation et cetera. Information-centric Networking (ICN) is a concept that changes the internet design which focuses on transmitting and accessing named content instead of the traditional host-based IP data. Several challenges present itself when we consider the dynamic infrastucture and deployment of IoT environment such as autoconfiguration of the names,enabling push traffic and minimising header size. However, the most significant chanllenge is to ensure the security and privacy of sensitive data transferred over the network.This report discusses the ICN-Iot middleware which provides a design level solution to the challenge of security and privacy in communication of data.**

I. Introduction

The Internet of things (IoT) is increasingly becoming part of our everyday life. In order to protect the processed data and the privacy of the users, it is important to implement dedicated security measures in IoT devices and networks.

Information-centric networking (ICN) is an approach to evolve the Internet infrastructure by introducing uniquely named data as a core Internet principle. Data becomes independent from location, application, storage, and means of transportation, enabling in-network caching and replication. The expected benefits are improved efficiency, better scalability with respect to information/bandwidth demand and better robustness in challenging communication scenarios.

ICN offers many features including caching, computing and storage, mobility, context-aware networking and support for ad hoc networking features, all of which have to be realized in an application-specific means in the context of IP-IoT. These compelling features enable a distributed and intelligent data distribution platform to support heterogeneous IoT services with features like device bootstrapping with minimal configuration, simpler protocols to aid self-organizing among the IoT elements, natural support for compute and caching logic at strategic points in the network.

Since data in information-centric networking is delocalized and need not be retrieved via an end-to-end transport stream but instead with hop-wise replication and in-network caching, it facilitates information dissemination in the IoT environment, and relaxs the demand for continued connectivity.

ICN concepts can be applied to different layers of the protocol stack: name-based data access can be implemented on top of the existing IP infrastructure, e.g., by providing resource naming, ubiquitous caching and corresponding transport services, or it can be seen as a packet-level internetworking technology that would cause fundamental changes to Internet routing and forwarding.

With ICN, new security models are needed. Today's host-centric trust model - retrieving data from a trusted server via a secure connection - no longer applies. Instead, security/trust/identity functions are bound to the information objects themselves, employing signed objects and and ensuring name-data integrity. Moreover, not all objects will be universally accessible, requiring authorization and scoping mechanisms.

The heterogeneity of both network equipment deployed and services offered by IoT networks leads to a large variety of data, services and devices. While using a traditional host-centric architecture, only devices or their network interfaces are named at the network level, leaving to the application layer the task to name data and services. This causes different applications to use different naming schemes, and no consistent mapping from application layer names to network names exist. In many common applications of IoT networks, data and services are the main goal, and ICN provides an intuitive way to name those in a way that can be utilized on the network layer as well. Communication with a specific device is often secondary, but when needed, the same ICN naming mechanisms can be used. The network distributes content and provides a service, instead of only sending data between two named devices. In this context, data content and services can be provided by several devices, or group of devices, hence naming data and services is often more important than naming the devices. This naming mechanism also enables self-configuration of the IoT system.

ICN provides data integrity through Name-Data Integrity, i.e., the guarantee that the given data corresponds to the name with which it was addressed. Signature-based schemes can additionally provide data authenticity, meaning establishing the origin, or provenance, of the data, for example, by cryptographically linking a data object to the identity of a publisher. Confidentiality can be handled on a per object basis based on keys established at the application level. All of this means that the actual transmission of data does not have to be secured as the same security mechanisms protect the data after generation until consumed by a client, regardless of whether it is in transit over a communication channel or stored in an intermediate cache. In an ICN network, each individual object within a stream of immutable objects could potentially be retrieved from a cache in a different location. Having a trust relationship with each of these different caches is not realistic. Through Name-Data Integrity, ICN automatically guarantees data integrity to the requester regardless of the location from where it is delivered. The Object Security model also ensures that the content is readily available in a secure state in the device constraints are severe enough that it is not able to perform the required cryptographic operations for Object Security, it may be possible to offload this operation to a trusted gateway to which only a single secure channel needs to be established.

II. Networking in IoT

A. The Protocol stack in ICN

One of the foundations of software engineering is the insatiable desire to recognize patterns exhibited by software and to collate this functionality into a abstractions accessible through some API(s). The TCP/IP network stack is a prime example of how these types of abstractions can be layered upon one another to provide a simple interface to complex machinery.

ICN would provide an abstraction in the form of named content rather than addressable end-hosts in the network. Along the TCP/IP stack, protocols and applications built on top became entrenched in APIs that were host-centric. HTTP, for example, interacts with resources from a specific server. When HTTP became the standard for web communication, the layers underneath were almost static.Beyond basic end-to-end connectivity, the network stack must deal with security and privacy via TLS, DTLS, or QUIC, transporting data, and managing session logic.

The traditional TCP/IP stack and the features it provides. Fig. 1.

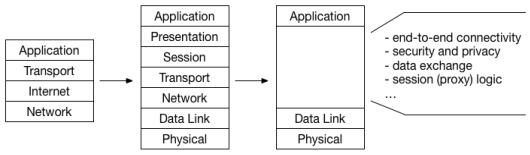

Fig. 1. TCP/IP stack

ICN tries to reallocate the functionality needed to get data from the network into sensible abstractions that better suit today's applications. ICN offers a compelling alternative to networking that might help us chip away at the technical debt that has accrued as our dependence on the TCP/IP model grew. This is the protocol stack of the new internet paradigm ICN. Fig. 2.

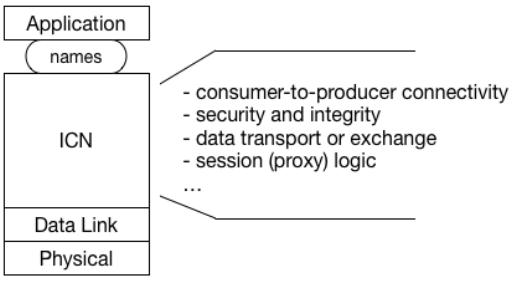

Fig. 2. ICN stack

B. IoT Middleware architecture

The unified IoT architecture, in which the middleware IoT services are proposed for administrative purposes such as towards onboarding devices, device/service discovery and naming. Other functions such as name publishing, discovery, and delivery of the IoT data/services is directly implemented in the ICN network. The proposed ICN-IoT unified architecture. Fig. 3.

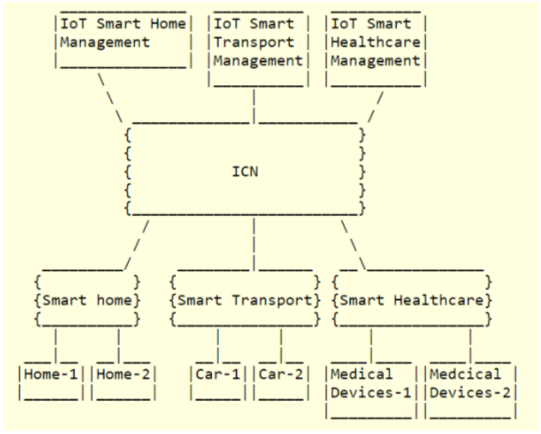

Fig. 3. ICN-IoT unified architecture

The five components of the architecture:

1) Embedded system
2) Aggregator
3) Local Service Gateway(LSG)
4) ICN-IoT Server
5) Service/Consumer

Here is the middleware Fig. 4.

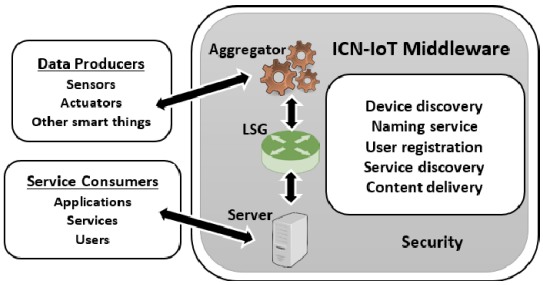

Fig. 4. Middleware architecture

Embedded Systems (ES): The embedded sensor has sensing and actuating functions and may also be able to relay data for other sensors to the Aggregator, through wireless or wired links. Aggregator: It interconnects various entities in a local IoT network. Aggregators serve the following functionalities: device discovery, service discovery, and name assignment. Aggregators can communication with each other directly or through the local service gateway.

Local Service Gateway (LSG): A LSG serves the following functionalities: (1) it is at the administrative boundary, such as, the home or an enterprise, connecting the local IoT system to the rest of the global IoT system, (2) it serves to assign ICN names to local sensors, (3) it enforces data access policies for local IoT devices, and (4) it runs context processing services to generate information specified by application-specific contexts (instead of raw data) to the IoT server.

ICN-IoT Server: Within a given IoT service context, the IoT server is a centralized server that maintains subscription memberships and provides the lookup service for subscribers.

Unlike legacy IoT servers that are involved in the data path from publishers to subscribers – raising the concern of its interfaces being a bottleneck – the IoT server in our architecture is only involved in the control path where publishers and subscribers exchange their names, certificates, and impose other security functions such as access control.

Authentication Manager (AM): The authentication manager serves to enable authentication of the embedded devices when they are onboarded in the network and also if their identities need to be validated at the overall system level. The authentication manager may be co-resident with the LSG or the IoT server, but it can also be standalone inside the administrative boundary or outside it.

Services/Consumer: These are other application instances interacting with the IoT server to fetch or be notified of anything of interest within the scope of the IoT service.

III. Secure Functionalities

A. Secure device Discovery

Device discovery serves two functions: 1) it is used in the context of discovering neighboring Embedded systems to form routing paths, where existing mechanims can be use; 2) for device onboarding, on which we focus here. During onboarding, the Embedded system passes its device-level information (such as manufacturer-ID and model number) and application-level information (such as service type and data type) to the upstream devices. But, if the device is required to have a globally unique ICN ID, it can be provided one by the naming service.

n the name-based ICN approaches where device identity is not needed, a device can publish within the name scope of the aggregator.In most IoT systems, devices interact with the aggregator for data or information processing or aggregation, hence there is no direct communication between devices under an aggregator. If in some set-up devices under the aggregator need to communicate with each other a scalable mechanism is to allow direct neighbors to communicate with each other while others communicate through the aggregator.

Fig. 5.

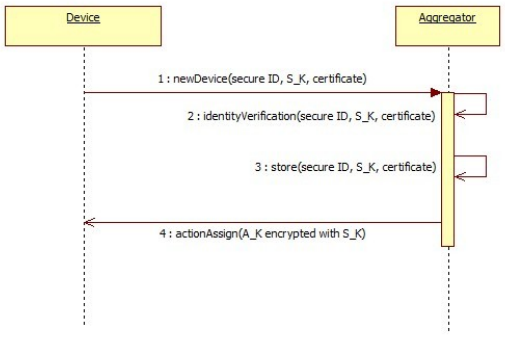

Fig. 5. Secure device Discovery

B. Secure service Discovery

Secure service Discovery Fig. 6.

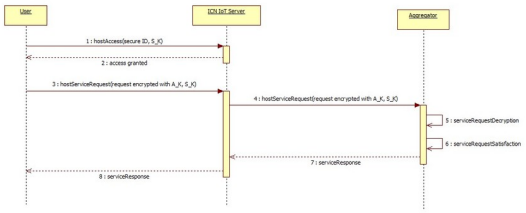

Fig. 6. Secure service Discovery

C. Secure naming service

If a device needs a global unique name/ID, but does not have one, it may request the naming service to obtain one after it is authenticated. Alternatively, the IoT domain (LSG or aggregator) may determine ID (name) for an authenticated device is required based on the policy. The proposed naming process works as follows. After a device has been authenticated, it may request an ID from the naming service (or the aggregator, if it can give the device a locally unique name). It sends a ID request (IDReq) to the naming service or aggregator. If the aggregator can accept request to give a unique name to the device, it will do that.

Fig. 7.

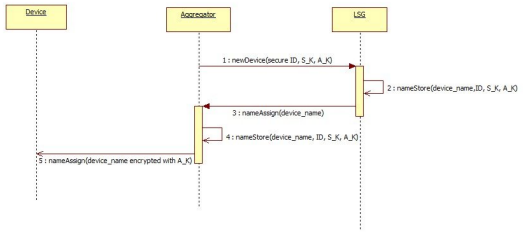

Fig. 7. Secure naming service

D. User registration

User Registration. A user, who wants to access/subscribe to a service, has to perform the registration operation by sending information that depends on the specific application domain to the IoT server. The information can be secured with the help of the PKI infrastructure. Upon successful registration the IoT server securely transmits an identifier, a user signature key SK (to be used to sign messages), a user encryption key EK (to communicate data confidentially), and an access password to the user in an encrypted message. Upon reception of the message, the user accesses the system to modify his/her password (function changePassword). With respect to existing secure application-layer solutions, a further benefit of the presented approach is the introduction of a second level of security, represented by the use of a temporary password (immediately replaced) and a couple of keys (signature SK and encryption EK), which is well suited for the heterogeneous and distributed IoT environment. Fig. 8.

E. Secure content discovery

Secure content discovery Fig. 9.

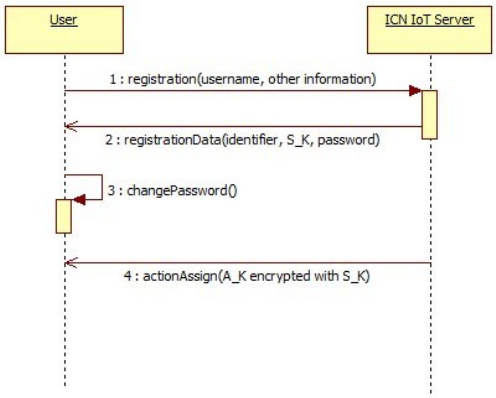

Fig. 8. User registration

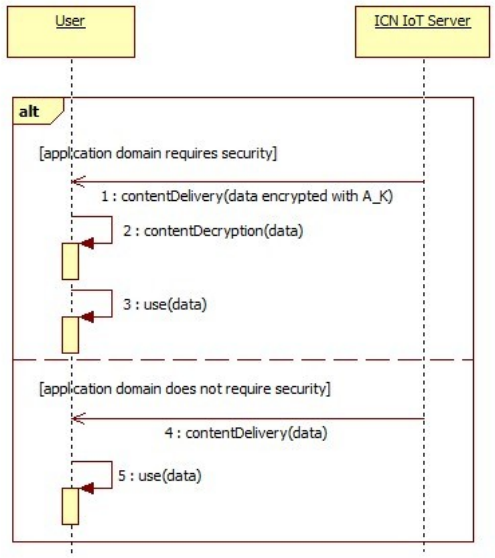

Fig. 9. Secure content discovery

IV. Discussion

There are ccertain open research problems in the IoT-ICN application:

1) How do applications get permission to publish and operate under a given name?
2) How are namespace collisions handled?
3) Can we break our dependence on a centralized or decentralized PKI?
4) How can we adopt distributed trust models?
5) How can we enable object encryption without sacrificing privacy ?
6) Can we build better broadcast encryption schemes that can handle the foreseen IoT scale?
7) How do we create secure routing hints?
8) Can we decouple “application names” from “network names”?

V. Conclusions

For the last 35 years, the current internet system struggles with enabling a secure communication between hosts. It is therefore of utmost imprtance to ensure a proper security and privacy measures to be introduced in the early stages of network design. The ICN-IoT middleware architecture is a propasal that will ensure the security functionalities required for secure communication.

References

[1] Why IoT with ICN?. http://conferences2.sigcomm.org/acm-icn/2017/files/tutorial-ndn-ccnlite-riot/2-Why-ICN-for-IoT.pdf.
[2] ICN based Architecture for IoT. https://tools.ietf.org/id/draft-zhang-icnrg-icniot-architecture-01.html.
[3] Why Bother with ICN?. http://chris-wood.github.io/2016/02/05/Why-ICN.html.
[4] Sabrina Sicari, Alessandra Rizzardi, Luigi Alfredo Grieco, Alberto Coen-Porisini. A Secure ICN-IoT Architecture. *IEEE International Conference on Communications Workshops (ICC Workshops)*, 2017.
[5] Mauro Conti,Giorgio Di Natale,Annelie Heuser,Thomas Pöppelmann,Nele Mentens. Do we need a holistic approach for the design of secure IoT systems?. *Proceedings of the Computing Frontiers Conference*, 2017.
[6] Emmanuel Baccelli,Christian Mehlis,Oliver Hahm,Thomas C. Schmidt,Matthias Wählisch. Information centric networking in the IoT: experiments with NDN in the wild. *Proceedings of the 1st ACM Conference on Information-Centric Networking*, 2014.