A Multi-Domain Framework to Enable Privacy for Aggregated IoT Streams

Renato Caminha Juacaba Neto, Pascal Mérindol and Fabrice Theoleyre
ICube, CNRS / University of Strasbourg, France
{caminha,merindol,theoleyre}@unistra.fr

2020 IEEE 45th Conference on Local Computer Networks (LCN) | 978-1-7281-7158-6/20/$31.00 ©2020 IEEE | DOI: 10.1109/LCN48667.2020.9314825

***Abstract*—The Internet of Things (IoT) is expected to integrate a large number of sensors, and actuators to the Internet. Multiple concurrent applications may cohabit on top of the same IoT infrastructure, and may re-use the same data for various purpose. However, privacy represents a major concern for many IoT applications, such as in smart building and healthcare. We propose here a multi-domain IoT framework where each domain aggregates distinct data-streams to respect their privacy concerns. We argue that removing sensitive meta-data and aggregating values reported by each data-stream is sufficient to hide individual private measurements. Moreover, relying on the Named Data Networking (NDN) paradigm, we can exploit caching strategies and perform in-network processing to ensure both scalability and privacy. In this paper, we discuss the necessary mechanisms to design a scalable inter-domain, privacy aware NDN scheme.**

I. Introduction

The Internet of Things (IoT) generates nowadays a significant volume of data. Most typical IoT applications rely on data-streams. Such streams are produced by sensors and then transmitted to controllers (consumers) that process the collected information to take smart decisions and possibly using actuators. For instance, e-health relies on a collection of sensors that monitor a large set of measurements (heart rate, movement, etc.). In the same way, smart meters measure the electricity consumption that is reported to the electricity provider to enable the smart grid. Since IoT streams can account for a large volume of data, there exists an opportunity to filter and aggregate them directly in the network to mitigate the overall network load while still offering the same application benefits to consumers.

To enable privacy, we propose to create a multi-domain IoT: while different owners can cohabit in the same shared network infrastructure, they should decide on their own the data they want to share. Thus, we should be able to filter the data-streams generated inside a domain, to avoid privacy leaks when streams are forwarded to a neighboring domain. In particular, we make a clear distinction between:

intra-domain streams, where privacy is not an issue: all the devices have the same owner;

inter-domain streams, where data should be filtered and aggregated to respect privacy concerns.

Current privacy enabling solutions often rely on ciphering mechanisms so that private data is only decodable by the consumers. An access control strategy is then required to regulate finely which entity can access to which data [1]. Unfortunately, maintaining the keys and consistent access control rules is very challenging in multi-tenant applications.

In this paper, we aim rather to enable privacy by design in multi-domain IoT. In particular, we look for a framework enforcing the respect of privacy constraints in a scalable manner. Data-streams are processed (*i.e.*, agregated and transformed) as they travel between domains. In summary, we propose two main contributions:

1) we introduce the requirements for multi-domain applications, and explain how Named Data Networking (NDN) can bring scalable privacy that is a major concern for a large-scale IoT deployment;
2) We detail how aggregation helps to reach this goal, while also allowing to benefit from the NDN caching strategy to save bandwidth and enable re-usability.

II. Related Work

Considering a decentralized IoT model requires both to respect privacy concerns, and to enable a large-scale inter-domain architecture.

A. Privacy enforcement

Privacy has always been a major requirement in IoT, particularly for sensitive information (*e.g.*, healthcare, personal sensors, smart buildings). However solutions must beware of the limited resources of IoT devices since costly operations jeopardize battery life due to longer processing and extra network exchanges [2].

By using attribute based access control, cipher keys are distributed by a mediator that verifies if the attributes of the consumer match the requirements of the producer [1]. Such approach requires several exchanges with the mediator in order to acquire the access keys before each data acquisition which can be quite costly to some IoT devices.

Anonymization, aggregation and filtering are alternatives to ciphering. They provide privacy by decreasing the precision of data. Aggregating data from several sensors hides the specific values of each sensor while providing information on the global population [3]. Filtering and anonymization remove or mask sensitive information from data until some quantifiable privacy requirements are reached, such as k-anonymity [4]. However, directly handling these filtering rules inside the network infrastructure is still a challenge.

B. Inter-IoT communications

Interoperability in the network stack is a requirement to allow several autonomous IoT domains to exchange data. Typically, proxies may help to translate information between different applications [5], or even aggregate data among different domains. They can also unify the semantics of data from different networks [6]. However, these proxies solely enable inter-domain data exchange, they do not handle privacy natively.

Instead, middle boxes may help to provide privacy at the borders of domains. Firewall-like devices are placed between domains to block messages that breach security policies [7]. However, the rules to apply on such devices are challenging to deploy and are only tailored for specific protocols.

III. Requirements to support privacy in multi-domain applications

We consider scenarios and applications possibly relying on large scale multi-domain IoT topologies. As such, we rely on a broad enough definition for describing such domains. They may either represent a limited collection of devices having the same owner, or having a specific application usage in common. In any cases, the key aspect is that the raw data may be exchanged within a domain, but privacy constraints hold when data exit each domain.

For example, in the context of smart buildings, the following properties illustrate well how a domain can be defined in practice:

Geographic: devices which are located within the same room / building / block;

Application-based: devices in charge of a given system. For instance, it can be a heating system regrouping temperature sensors, electric boiler, valves for the radiators and the controller;

Manufacturer: devices of the vendor providing statistics usage or predictive maintenance.

Finally, it is worth noting that we do not exclude the case of a device belonging (directly or indirectly) to multiple domains, *e.g.*, the same sensor can re-used by several local applications and is monitored remotely by the manufacturer.

A. Objectives

Here is the general challenge we aim to solve:

*How to **efficiently** collect and aggregate/combine numerous **data-streams** from multiple **independent domains** while preserving the **privacy constraints** of each data-producer?*

This question can be split into four specific objectives that are the underlying guidelines of our solution:

G1 Support heterogeneous applications with multiple independent producers;
G2 Maintain producer-specific privacy constraints;
G3 Enable large scale data-stream exchanges;
G4 Answer IoT queries efficiently.

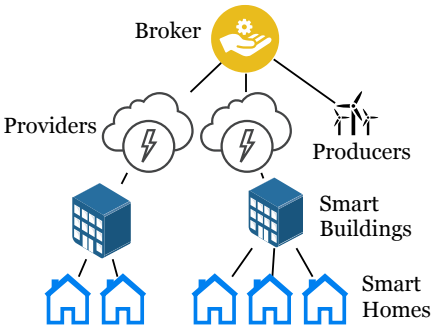


Fig. 1: A multi-domain scenario.

B. Multi-domain Illustration

Let us consider the scenario in Fig. 1 to illustrate our goals. Each smart home is typically an independent domain. These domains can interact with a larger scale domain, a smart building that aims to manage some characteristics of each smart home (*e.g.*, for the heating system). In turn, in order to implement the smart grid and provide dynamic pricing (goal **G1**), electricity providers need to get the overall real-time consumption of the buildings.

While *some* information need to be exchanged among the different domains, it is also critical to respect the privacy concerns of each smart home (goal **G2**). That is isolating a specific stream (from a given home) should not be possible for non trusted domains. On a large scale, an aggregated real-time information (with no privacy leak) is generally enough for companies optimizing their process. It also favors the potential emergence of brokers that can *sell* the aggregated data they collected, leading to a global interconnection (goal **G3**) with various consumers and producers. The main challenge being here to respect privacy concerns without compromising the efficiency of the network infrastructure. We aim to rely on caching strategies and in-network processing to safely re-use the same pre-processed data for different interests (goal **G4**).

C. A Query Model for Data-Streams

Typically, in our model, a request describes the desired data by its type (*e.g.*, temperature, humidity, wattage), cardinality (the number of samples and sensors), frequency and location. For example, a query may consist in asking for the humidity of soil in farms of a specific region, or in the power consumption from at least 100 houses every day. Such a description is referred as the metadata provided in the query.

When a domain receives a query, it has to find and process the data to match all the metadata criteria. As a typical request asks for an aggregated value (both spatially and in time), the domain handling it is both in charge of retrieving the data and applying such a transformation to provide the desired granularity to consumers. For instance, the energy consumption may be averaged monthly and this simple transformation should be applied as close as possible from the producer. Indeed, it is not only about scalability (by transmitting only the resulting value, the network load decreases), aggregations also enable

privacy by masking specific samples. When possible, a domain may apply such operations by itself or trust others to do it.

D. Trust Model between Domains

A producer may want to apply some critical pre-transformations on the data it owns to provide a minimum level of privacy (e.g. with anonymization or watermarking) before exchanging data with its peers. Then, this peering domain may also share the data not produced locally with others domains and so on. In that case, the producer may ask additional privacy constraints, such as k-anonymity with other domains. Similarly, if a producer blacklists a given domain, its peer must not export this data to the non trusted domain.

To exploit the full potential of our approach, we assume that peering domains trust each other, *i.e.*, one domain can expect its peers to respect the privacy policies they agree on. Tools like watermarking [8] may help producers to verify that their peers correctly respect the defined policies. When a leak is detected, watermarking indeed allows to identify the faulty peer in the chain. That is why we expect that watermarking may typically be part of the critical and minimal set of pre-transformations applied before sharing any data-stream.

IV. A NDN Multi-domain Architecture

In the IoT context we consider, applications generate *streams*, *e.g.*, chronological sequences of measurements, sent periodically. The NDN paradigm fits well with such applications because it can efficiently deal with IoT queries treated as interests. We adopt the following terminology to put the data at the core of our forwarding model:

- a **chunk of data** is defined as a piece of data (*e.g.*, sensor's measurement);
- a **data-stream** is a temporal sequence of chunks;
- a **dataset** represents the data that a domain accepts to export, *i.e.*, the values and its semantic characteristics like the nature and cardinality of the dataset (*e.g.*, temperature measurements from 1,000 sensors).

A. NDN Support of IoT streams

NDN matches the design and needs of most IoT applications. By forwarding interests and datasets, routers directly manipulate chunks of data, and not anymore opaque packets. Hierarchical names replace numerical addresses, and this naming hierarchy enables route aggregation thanks to prefix based routing. Each NDN router has a cache (a.k.a. *content store*) to maximize data re-usability, in particular for popular interests. The cache policy behaves in the following way: i) a router may insert in the content store any dataset that is locally generated or forwarded; ii) routers forward an interest if the answer is not in the content store, else a reply is directly sent to the inquirer.

However, some NDN features need to be adapted specifically for IoT needs [9]. In particular, to support IoT data-streams: sensors generate a sequence of measurements, that are exploited by consumers (*e.g.*, HVAC uses the last temperatures measured in a room). Besides, making the reverse path entries persistent allows the consumers to implement

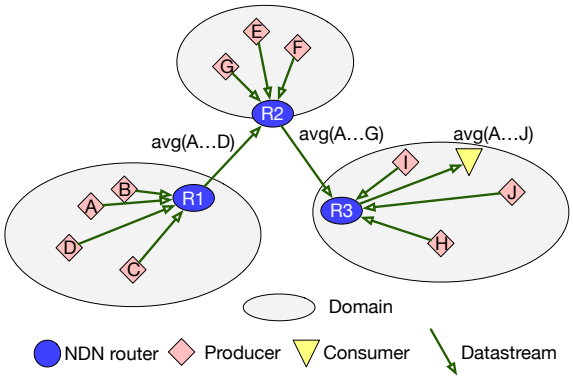


Fig. 2: A multi-domain NDN topology.

subscriptions [10]. Producers can then keep on pushing their measurements to their *subscribers*.

B. Towards an Inter-Domain Architecture

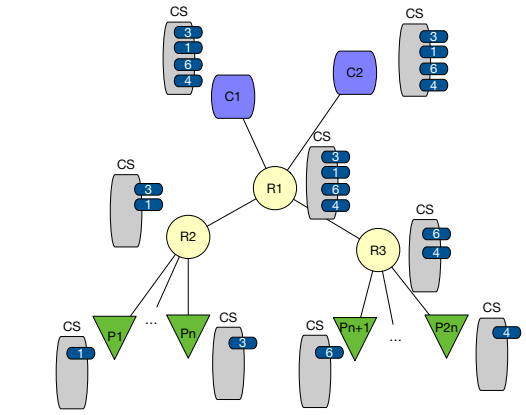
We envision a large scale NDN topology, where couples of peering IoT domains rely on producer-consumer relationships. More precisely, when datasets are exported, each NDN router is in charge of executing the privacy functions. Let us consider Fig. 2: R1 has to aggregate the 4 data-streams (from A, B, C, and D), giving only the average value over 4 samples. Then, this novel dataset is consumed by R2, that aggregates it with the local streams (from G, E, and F) to form yet another dataset that is used by R3 with its local streams (from I, J, and H) to answer the final consumer.

This aggregation, performed at each hop of the inter-domain route, helps to respect the privacy concerns as it hides individual streams. To enable such a feature, NDN routers expose to their peers the dataset they aim to export (using the metadata description). When datasets can be aggregated, because they share the same nature, a novel (super) dataset can be exported in its turn. As any device (in any domain) can query any dataset, being generated locally, aggregated or exported from peering domains, this leads to an inter-domain architecture enabling privacy by design.

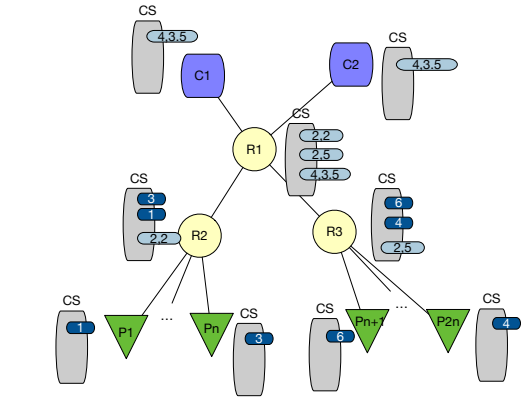
C. Aggregation and Filtering in NDN

Ciphering each chunk of data is not scalable: the producer must know a priori all the consumers which are susceptible to query its data. Rather we propose to integrate aggregation and filtering features directly in the NDN routers. For instance, we propose to remove sensible attributes, average data from multiple producers, or decrease the accuracy of some descriptors. It strongly mitigates the ability to identify a given producer and exact values. In particular, there exists some privacy metrics that allow to finely tune and quantify the achieved level of privacy of such general methods. For example, K-Anonymization denominates that each sample is indistinguishable from $K-1$ others while ϵ-differential enforces that several datasets are similar enough with each other.

In our architecture, the role of a NDN router is to apply such transformations on the exported datasets. They are in charge of



(a) Conventional flat NDN.



(b) Aggregation based inter-domain routing.

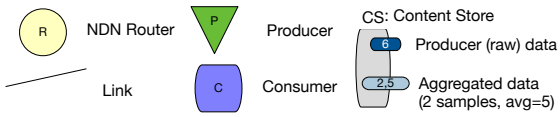


Fig. 3: Topology with 2 consumers and $2 \times n$ producers.

constructing novel datasets, from the locally produced streams and imported ones. These dataset should comply to a set of privacy concerns that are configured with both local and peering policies. These policies typically take into account the cardinality on which the aggregation function should apply (*e.g.*, an average value over at least 20 producers) for a given type of measurements.

V. Illustrating the Power of Aggregation

Let us consider a simple scenario to compare a flat approach (Fig. 3a) to our inter-domain aggregation model (Fig. 3b).

In a flat NDN approach, each producer sends its raw stream directly to the consumers. This strategy is efficient only if several consumers (here C1 and C2) are interested in the same data. An intermediary NDN router (*e.g.*, R1) may have already the popular chunk of data in its cache to reduce the bandwidth consumption. However, each consumer needs to apply by itself the aggregation function relevant for its application. Raw data are disseminated in all the infrastructure, which is prejudicial to privacy if not ciphered.

On the contrary, our aggregation-based strategy processes the datasets directly inside the network. For instance, R2 is both a consumer and a router as it consumes the streams from its domain, and generates the average value (2) over 2 samples; this dataset is then re-exported to R1. R3 behaves similarly, only exporting aggregated data. As with the flat approach, the data can also be cached efficiently, but here Content Stores contain only the aggregated values, not each sample individually. It does not only bring scalability but also privacy: final consumers never access and process the individual raw data on their own.

VI. Conclusion

In this paper, we presented an NDN multi-domain architecture for IoT streams. As privacy represents a major concern, we propose that each NDN router is in charge of filtering and modifying the data when it exits its domain. In particular, with data aggregation, a consumer cannot infer individual measurements from the aggregated one. We argue that processing the streams directly in the network conforms to the requirements of many IoT applications. Besides, by reusing the NDN caching strategy to store aggregated chunks of data, a router can use its Content Store to serve different consumers that have similar interests. As future works, we plan to investigate the complete integration of these features in a multi-domain routing algorithm designed for NDN.

Acknowledgment

This work was supported by the French National Research Agency (ANR) project Nano-Net under contract ANR-18-CE25-0003.

References

[1] B. Li, D. Huang, Z. Wang, and Y. Zhu, "Attribute-based Access Control for ICN Naming Scheme," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 2, pp. 194–206, mar 2018.

[2] J. Lopez, R. Rios, F. Bao, and G. Wang, "Evolving privacy: From sensors to the Internet of Things," *Future Generation Computer Systems*, vol. 75, pp. 46–57, 2017.

[3] I. Wagner and D. Eckhoff, "Technical Privacy Metrics: A Systematic Survey," *ACM Computing Surveys*, vol. 51, no. 3, pp. 57:1–57:38, 2018.

[4] L. SWEENEY, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[5] A. P. Castellani, T. Fossati, and S. Loreto, "Http-coap cross protocol proxy: an implementation viewpoint," in *MASS*. IEEE, 2012.

[6] P. Desai, A. Sheth, and P. Anantharam, "Semantic Gateway as a Service Architecture for IoT Interoperability," in *International Conference on Mobile Services*, vol. 32, no. 2. IEEE, jun 2015, pp. 313–319.

[7] L. Metongnon, R. Sadre, and E. C. Ezin, "Distributed Middlebox Architecture for IoT Protection," in *CNSM*. IEEE, oct 2019.

[8] F. Cayre, C. Fontaine, and T. Furon, "Watermarking security: theory and practice," *IEEE Transactions on Signal Processing*, vol. 53, no. 10, pp. 3976–3987, 2005.

[9] R. Ravindran, Y. Zhang, L. Grieco, A. Lindgren, J. Burke, B. Ahlgren, and A. Azgin, "Design Considerations for Applying ICN to IoT," ICN Research Group, Tech. Rep., 2019.

[10] R. C. Sofia and P. M. Mendes, "An Overview on Push-Based Communication Models for Information-Centric Networking," *Future Internet*, vol. 11, no. 3, p. 74, mar 2019.