Transport Layer Efficiency and Security in IoT: RINA's Approach

Peyman Teymoori and Toktam Ramezanifarkhani
Department of Informatics
University of Oslo, Norway
Email: {peymant|toktamr}@ifi.uio.no

***Abstract*—Technological advancements have enabled us to connect tiny things to the Internet to benefit from an automatized, easier life style using the Internet of Things (IoT). This, however, raised serious challenges, especially at the communication/networking layer of IoT. In particular, meeting both of the transport layer efficiency and security has been a serious research question and open problem. In this paper, we take a deeper look at this problem by summarizing current network stacks of IoT, and then, through some use cases, we demonstrate how the idea of recursion in networking can solve this problem; we employ the Recursive InterNetwork Architecture (RINA) as our architectural approach, and show how this idea can conquer the above challenge without sacrificing security or efficiency, and how secure multicast, a crucial requirement of IoT, can be inherently performed in IoT.**

I. Introduction

The IoT aims to provide a communication network infrastructure with inter-operable protocols and software for physical/virtual sensors, personal computers (PCs), smart devices, automobiles, and other items, such as a refrigerator, dishwasher, microwave oven, food, and medicines, anytime and on any network. Since IoT embraces many different technologies, services, and standards, it is widely perceived as a main pillar of the ICT market in the next ten years [1]. Many new applications such as health-care (e.g. remote patient/elderly people monitoring) and home automation (e.g. heating and lightning control) have been developed so far. This has led to efforts conducted by standardization bodies such as the Institute of Electrical and Electronics Engineers (IEEE) and the Internet Engineering Task Force (IETF) [2], towards the design of communication and security technologies for the IoT. Given the availability of wireless communication technologies, the number of developed things is expected to rapidly increase: this also raises serious challenges: the network can be easily targeted by attacks and malicious users. What IoT needs is a well-established efficient communication platform across different domains, capable of ensuring the security of information.

Although most of these issues have been found and tried to be addressed before in the Internet, however, IoT still faces almost the same issues; to improve the efficiency at the transport layer, performance enhancement methods such as PEPs and CoAP gateways have been proposed in IoT. DTLS, which is based on TLS but operating on datagrams, has been adopted by IoT to address its security challenges. Many IoT protocols such as MQTT and CoAP utilize TLS or DTLS. However, the performance of CoAP proxies is affected inversely by adding security features to other layers especially, transport. The main reasons include [3] 1) limitations of the DTLS Handshake Protocol with large messages, 2) complications of using DTLS with CoAP, and 3) not supporting multicast communications by DTLS. In other words, adding (security) functionalities to one layer might interfere proper (and mostly performance) operations at the other layers [4]. The problem worsens when an IoT application is supposed to operate over multiple cross-domain nodes and heterogeneous environments; this is the cause of diversity of protocols and policies and their communication, especially from the security point of view.

In this paper, we show that the contradiction between transport layer efficiency and security arises from the architectural deficiency of IoT, and to demonstrate how this problem can be solved, we present an architectural solution based on the idea of recursiveness in networking functionality. In particular, we focus on the Recursive InterNetwork Architecture (RINA), and show how this architecture can meet both efficiency and security, and how recursiveness can facilitate providing a unified interoperable and programmable architecture. RINA was first introduced in [5] and then, implemented and evaluated by a number of international projects. Recursion in RINA is performed by defining a layer as a foundation with basic *mechanisms* to provide Inter Process Communication (IPC) between two IPC Processes (IPCP). This layer is called DIF (Distributed IPC Facility). DIFs (the mechanisms inside DIFs) can be customized through adopting different *policies*. Then, DIFs can be recursively arranged to form different topologies [6]. The recursion idea and using fixed mechanisms at every layer reduces the number of protocols and security challenges at different network layers since the same program is instantiated. The recursive idea of RINA enables us to solve the aforementioned problems of IoT regarding its transport efficiency and security.

The paper is organized as follow: in Section II we discuss current problems IoT is facing in securing, especially at its transport layer. Section III overviews RINA with a focus on its transport features. We demonstrate how recursion in RINA can solve the problem in Section IV. Section V discusses further research topics, and we conclude in Section VI.

ISBN 978-3-903176-28-7

II. The Internet of Things (IoT) Network Stacks

IoT needs is an efficient communication platform across different domains, which is capable of ensuring security. We argue that IoT security challenges mostly stem from the architectural design of the IoT network stack. There are two tracks of efforts on improving the security in IoT. One of them utilizes the current Internet design (i.e. TCP/IP) as the base, and the other adopts new frameworks such as Information-Centric Networking (ICN).

A. IP-Based Stack

As IoT envisions a future Internet in which everyday objects possessing sensing and actuating capabilities cooperate with computer systems, IP-based communication protocols have been aligned with IoT devices to form a communications stack. A typical network stack of IoT with common protocols in each layer resembles the TCP/IP stack with some commonalities, especially in the design. The protocols developed by the Internet Engineering Task Force (IETF) working groups on IoT include 6LoWPAN, ROLL, and CoRE, which have been trying to use IP as the network infrastructure. However, the large-scale deployment of IP-based IoT solutions is still challenging [7]. In [8], it has shown that IoT security challenges of IP-based IoT frameworks mostly stem from the architectural design of the IoT network stack. In addition, comprehensive studies such as [9] on IoT security have confirmed that many issues stem from the currently-employed network stack protocols of IoT devices and their interoperability issues with the Internet. There some major flaws with IP: the concept of scope of a layer is not used correctly [10], as it has been diluted by gradual updates. Although some research work (e.g. [11]) focused on presenting a secure IoT architecture, it usually operates at higher layers regardless of what the network stack is. On the contrary, in this paper, we fundamentally look at the lower layers, and especially the stack and its protocols.

At the transport/application layer, DTLS, as the common protocol, have been in use to ensure end-to-end security using CoAP. As its limitations were mentioned by [3], [4], DTLS does not support multicast communication which is highly required in IoT networks. Due to its end-to-end security, DTLS also complicates operations of CoAP proxies in the path. This problem has led to securing CoAP communications and object security rather than the transport security provided by DTLS. However, this approach is not mature yet. Although most of these issues have been found and tried to be addressed before in the Internet, however, IoT still faces almost the same issues.

B. ICN-Based Stack

Different from the IP network, Information-Centric Networking (ICN) assigns unique names to content, regardless of its location (e.g. IP address) and generating application, which enables in-network caching/replication and content-based security [7]. Although the early ICN architects did not have IoT in mind, ICN has been adopted to address IoT requirements. However, the uniqueness and complexity of IoT requirements raise challenges to the design of ICN [7]. Despite the standardization efforts on ICN-IoT deployments within the ICN Research Group (ICNRG) [12], it has been under debate in the ICN community that ICN can face serious security and performance issues in IoT [7], [13] since ICN-IoT not only connects many more devices than most other networks, but it also assigns (unique) names to their content.

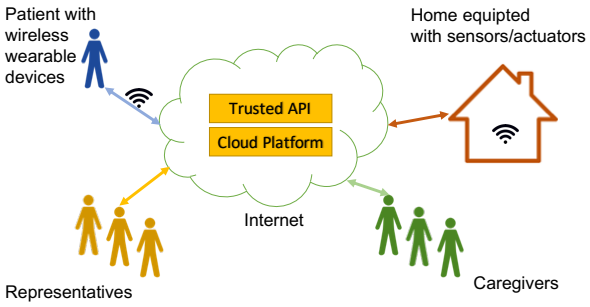

Fig. 1. Different IoT domains: health and home.

C. Challenges

From the above discussion, we can infer that there is a trade-off between transport layer efficiency and security in IoT in the aforementioned network stacks. To illustrate the problem via an example, consider IoT applications such as home monitoring and e-health as shown in Fig. 1 with the goal of monitoring the health of patient/elderly at home, and in case of emergency, providing a cooperation between these two systems; the level of security (trust/privacy) between patient/elderly monitoring IoT devices, and the center in the hospital, and the trust between these systems and the IoT devices connected to home monitoring is different. In fact, these systems should be connected (e.g. in case of emergency, caregivers should be allowed to enter home). However, the trust inside and between home monitoring and e-health is dependent on many different and temporary situations. In addition, IoT devices do need the support of (CoAP) proxies to be able to talk to the IoT cloud and other systems via an HTTP-like protocol; these proxies might need to break an end-to-end connection or even translate the protocol to another one at the cloud side to efficiently utilize resources at both sides. These all make communications in the IoT very hard to secure if the focus is on performance, or inefficient if security has a high priority. The problem aggravates by the crucial need to a multicast protocol in IoT; although there are proposals for just multicast in IoT such as [14], securing multicast, however, cannot be currently performed by DTLS [3].

III. Recursive InterNetwork Architecture (RINA)

A. Introduction

The Recursive InterNetwork Architecture (RINA), presented by John Day in “Patterns in Network Architecture: A return to Fundamentals” [5], is a novel approach in networking architecture trying to avoid architectural problems of TCP/IP and other technologies. Through evaluating it in different use cases (e.g. see (e.g. [6], [10], [15]–[18])), this is shown

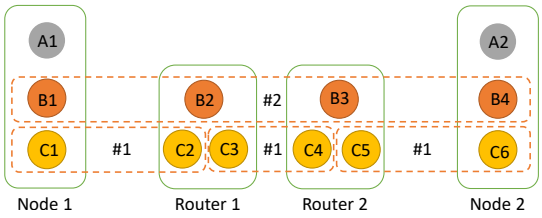

Fig. 2. A sample topology represented by DIFs.

that RINA can provide significant benefits to networking by removing architectural obstacles[1].

As shown in Fig. 2, RINA is based on a single type of layer (represented by dashed boxes in the figure), which is repeated as many times as required by the network designer. The layer is called a Distributed IPC Facility (DIF), which is a distributed application that provides Inter Process Communication (IPC) services over a given scope to the distributed applications above (which can be other DIFs or regular applications). These IPC services are defined by the DIF API. All DIFs offer the same services through their API and have the same components and structure. However, not all the DIFs operate over the same scope and environment nor do they have to provide the same level of service. In RINA, a Distributed Application is a set of two or more Application Processes (APs) that cooperate to do some function. The set of these APs is called a Distributed Application Facility (DAF). DAFs use DIFs when IPC between the APs in DAF is not possible via shared memory. In RINA, invariant parts (mechanisms) and variant parts (policies) are separated in different components of the architecture. This makes it possible to customize the behavior of a DIF to optimally operate in a certain environment with a set of policies for that environment instead of the traditional "one size fits all" approach or having to re-implement mechanisms over and over again.

In Fig. 2, there are four devices: Node1 and Node2 are two end-hosts that are connected via two routers. A1 and A2 are two application processes willing to communicate, and the circles in the DIFs are called IPC Processes (IPCPs). IPCPs have the same structure that can join DIFs. DIF #2 spans the whole network, allowing packets to be routed and congestion-controller from A1 to A2 via routers. There are also three DIFs #1 (could be assumed as the link layer) connecting devices together (doing mostly flow control). However, the structure of the DIFs are the same, and just the scope of the DIFs is different. This solves the problem of inventing new layers (as those presented by MPLS, VLAN, tunneling, etc.) and their deployment; the same layer can be customized via policies and instantiated to operate over different scopes.

Fig. 3 illustrates the data transfer modules inside IPCPs in a DIF. These modules show in each DIF, there is one (EFCP) sender and one (EFPC) receiver. The IPCP in the middle routes PDUs received from the left IPCP to the right one via its corresponding port. This scheme repeats at different DIFs, with the difference that it could be longer or as short

[1]See [19] for a list of publications on RINA.

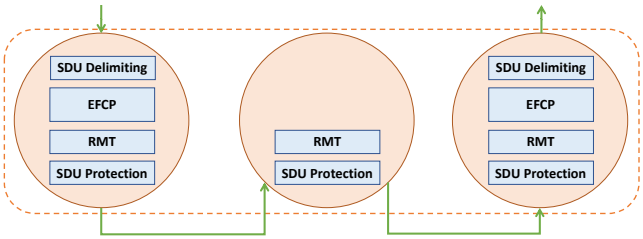

Fig. 3. Data transfer modules of three IPCPs inside one DIF.

as a point-to-point link without the routing IPCP. In [6], we evaluated the performance of a recursion of two layers of congestion controllers using a TCP Reno-like controller; we showed that the recursion can improve throughput while decreasing queuing delay compared to and end-to-end TCP Reno and Split TCP (or PEP), where a connection is broken at routers.

The main difference between RINA and other stacks is that RINA recurses the same layer, called DIF numbered from 1 (e.g. 1-DIF, 2-DIF) as the lowest one, and (N)-DIF as the current one we are focusing on. At the lowest layer, *shim* DIF operates over the physical layer, but it has the deployment possibility of operating on other protocols such as UDP.

B. IPCP's Data Transfer Mechanisms

A node joins a DIF through an IPC Process (IPCP), which is an instance of the same code handling IPC. IPCPs have the same structure consisting of the following mechanisms that operate at different timescales (ordered from faster to slower):

1) Data Transfer: handles packet transmission including:
   - Delimiting: encodes Service Data Units (SDUs) that arrive from the upper DIF within PDUs.
   - Error and Flow-Control Protocol (EFCP): handles data transmission using its two sub-protocols: Data Transfer Protocol (DTP) and Data Transfer Control Protocol (DTCP).
   - Relaying and Multiplexing Task (RMT): performs routing of PDUs to output ports of the DIF or upwards.
   - SDU Protection: performs encryption, compression, error-code and TTL.
2) Data Transfer Control: manages error, flow, and retransmission control.

C. Error and Flow-Control Protocol (EFCP)

EFCP is the only data transfer protocol in RINA, which is based on Richard Watson's fundamental results on synchronization for reliable data transfer [20]. Its main functions are sequencing, flow control, and retransmission control. EFCP provides an inter-process communication service to upper processes that might be an (N+1)-IPCP or and application process. The upper process is connected to (can write to/read from) the EFCP in the (N)-DIF via the (N)-port-id.

EFCP has two logical components operating at different time scales: 1) Data Transfer Protocol (DTP) that is the mandatory part of EFCP and includes tightly bound mechanisms, and 2) Data Transfer Control Protocol (DTCP), which include

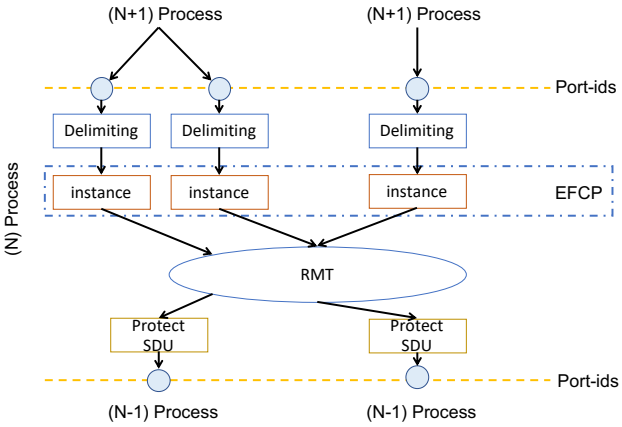

Fig. 4. A sample of data transfer modules.

loosely bound mechanisms. DTP is instantiated every time a flow is created. It is roughly similar to the UDP protocol. However, depending on the QoS requirements, DTCP might be also instantiated to provide retransmission and flow control.

Watson's result imply that the necessary and sufficient condition for synchronization to have a reliable data transfer is to set an upper bound on only these times: Maximum Packet Lifetime (MPL), maximum time to wait before sending an Ack, A, and the maximum time to exhaust retries, R [20]. This decouples port allocation and synchronization.

EFCP can be customized via several policies. A simple example is that the behavior of different TCP flavors can be imitated by writing different policies. The mechanisms operating on PDUs and policies are the same.

As shown in Fig. 4 , when a PDU is written to an (N)-port-id, the delimiting module, associated to the port, processes the PDU by possible fragmentation or concatenation. The result is passed to the EFCP instance associated to that port. As a result, generated PDUs are passed to the RMT for multiplexing. The RMT sends PDUs to one or more (N-1) ports. When the RMT reads a PDU from an (N-1)-port, if the PDU's destination address is not in another node, it passes the PDU to the relevant EFCP instance for further processing. EFCP instances of (N)-DIF are managed by the module Flow Allocator (FA) in that DIF. FA can create a new EFCP instance, replace an old instance by a new one, or delete the instances.

D. RINA's Transport Security Features

How RINA connects IPCPs and how their EFCP is connected to each other in different DIFs can provide other benefits than just performance. A complete evaluation is found in [8], [18]. Here, we summarize the security features related to EFCP:

1) Secure DIFs: "DIF is a secure container" [15]. This is the main feature that RINA secures layers instead of protocols. This means that all the packets leaving an (N)-DIF are protected via the SDU Protection module. Hence, IPCPs in (N-1)-DIF cannot inspect the arriving packet, and this can continue downwards in lower DIFs, which also depends on the policy of those DIFs. However, every IPCP in the same DIF can inspect the packet, and in case of performing enhancements or protocol conversion in CoAP proxies, packets are not obscured. This also implies that in RINA, IPCPs should be enrolled first before joining a DIF, and their access rights are validated. This also implies that an outsider cannot attack the DIF. In case an attacker can compromise the enrollment or if the DIF communication is not encrypted, RINA's typical field lengths in packets are still long enough to make attacks harder to succeed, e.g. 2^{48} possibilities to guess the connection information in RINA compared with 2^{29} possibilities in TCP during data transfer [15].

2) Hidden Addresses: RINA has a special addressing rule: IPCPs in each DIF have their own addresses, meaning that addresses are hidden from other DIFs. This can mitigate the problem that IP addresses are public in the Internet without any authentication.

3) Synchronization-Independent Port Allocation: In TCP, synchronization and port allocation are coupled, which makes it vulnerable. However, the decoupling of these two in RINA reduces the chance of intercepting a connection, and as a result, attacks are harder to mount.

4) Port-Independent Communication: In RINA, there is no well-known ports to listen to; applications are requested for service through their application name. RINA decouples port allocation and access control from data synchronization and transfer which makes attacks harder to mount [18].

5) Soft-State Connection Management: As stated before, RINA adopts Watson's method. This means that there is no explicit control messages for connection establishment/close. There are only timers, and a receiver deletes the state after 2∗MPL (Maximum Packet Lifetime). This reduces the chance of connection misuse [15].

6) Connection Management Independent Authentication: Since IPCPs are first enrolled in a DIF, authentication is not a part of connection management in RINA.

7) Variable Address Space: In addition to the addresses that are invisible to other DIFs, the size of addresses can vary in DIFs, which makes it harder for attackers. The address size, however, depends on the number of nodes in that DIF to save space in the packet header.

IV. Employing RINA for IoT Security

A. Use Cases

We consider two use cases to discuss how RINA and be applied on. In the first one, *Home*, we assume that IoT devices can run that RINA stack, and in the second one, *Health*, we assume that devices have their own legacy protocol.

1) Home: The building & home / smart infrastructure domain is usually covered by standard systems designed to manage building energy, environmental conditions and building security. The systems are usually provided by different vendors and can also be integrated together to provide the desired functionality. One of the major challenges here is the integration of applications that have a wide range of requirements, especially security.

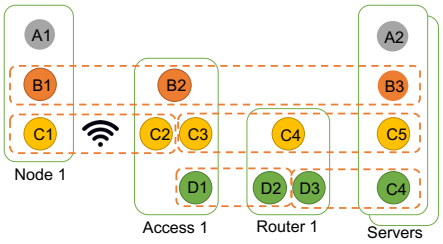

Fig. 5. DIF arrangements for the home use case.

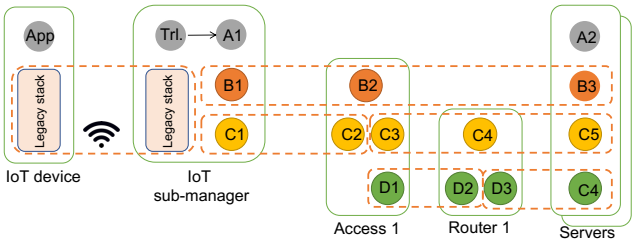

Fig. 6. DIF arrangements for the health use case.

Fig. 5 illustrates a customization of RINA for a simple, common IoT application: we assume that there are some wireless nodes connected to the network via an access point over a wireless protocol. This requires a common DIF between the two devices (with IPCPs C1 and C2). The wireless node is also a member of a DIF spanning over the network to reach the servers in the cloud. The communication, however, crosses a network segment (between C3 and C5) that is not a member of the top DIF, meaning that all the packets between B2 and B3 are (can be) out of access to Router1. This implies that the congestion controller between C3 and C5 can be customized to that segment while the congestion controller between C1 and C2 takes the wireless link into account; both path segments can have their own SDU protection policy, while B2 is able to perform any operation such as translation/conversion that is required on the PDU sent from B1 to reach B3.

2) Health: There is a need to provide timely assistance to people who run an increased risk of accidents, malfunctioning, diseases, strokes, etc., in particular elderly people, recovering or disabled people and possibly children. The goal is to monitor these people (target group) and their behavior by IoT wearable devices in order to reduce risks, to avoid accidents, to give preventive warnings, to initiate corrective measures or to ask for assistance, e.g., from nearby "trusted" caregivers.

A sample topology of a health IoT device that does not support RINA is shown in Fig. 6. As proposed by [21], legacy IoT devices can join a RINA network using a gateway called IoT sub-manager. We adopt the same approach here focusing more on the security and transport aspects of it. Referring to the figure, we observe that the network segment from IoT sub-manager to servers is the same as before; it seamlessly routes packets while considering their security without the need to an end-to-end connection. However, IoT sub-manager needs to translate the legacy protocol/packets from the IoT device, which is performed by the Trl module. Trl then passes data to A1, which is treated similarly to the data given to A1 in Fig. 5. This implies that IoT sub-manager should be a trusted device. There is another point in using IoT sub-managers: if the send rate of the IoT device is faster, i.e. there is a bottleneck in the RINA segment, Trl should be able to pushback/slow down App. This is, additionally, a part the translation Trl does.

B. Secure Multicast

In both of the above use cases, secure multicast can be performed without any challenges. Assume that the goal is to send data to several servers as shown in Fig. 5 and Fig. 6. This indeed does not need several separate connections from the IoT device to the servers. Instead, a multicast destination address is defined in DIF B. IPCP B1 sends the packet it gets from the IoT device to B2. Then, the RMT module in B2 checks the destination port(s); in this example, there are several servers connected to Access 1 via Router 1, each with a separate DIF with C3, C4, and C5. The RMT of B2 then copies the packet to all of these ports. As a result, IPCP C5 in all the servers receives the packet and forwards it upward. In this way, the security of the packets in DIF B is not compromised.

V. Discussion

A. Deployment

Similar to any new technology, RINA needs to be adopted gradually using different methods. RINA has the ability to operate over shim DIFs, which make it operational even on a TCP/IP stack. However, to fully benefit from RINA, some research work has been done on its deployment. For example, in [21], it has been shown how RINA can be used as an architecture in IoT when nodes cannot run RINA directly. Nodes, instead, connect to an IoT sub-manager, and the rest of the network can be a RINA network. In [22], the possibility of switching to RINA was evaluated in case a node can join a DIF directly via its first connected hop. In the RINAiSense project[2], the possibility of running RINA on limited devices such as sensors has been investigated. Deployment possibilities of RINA is also one of the main goals of OCARINA[3]. It was also shown that legacy applications that can use the TAPS API, can use RINA via a mapping of TAPS to RINA [23].

B. Further Research Topics

The ability of breaking an end-to-end connection without sacrificing security opens further research questions and challenges. In terms of performance, it is important to utilize an efficient feedback method in a recursive architecture in which, congestion controllers can be sequenced or stacked. In [24], we have analytically evaluated the effect of various feedback methods on the system's stability and average queue length; we found out that strict pushback feedback based on queue size can raise stability problems. However, designing a better method that does not rely on only the queue size is an interesting problem. We used Scalable TCP in [24]; this can, however, be improved by adopting a congestion controller

[2]See https://distrinet.cs.kuleuven.be/research/projects/RINAiSense
[3]See http://www.mn.uio.no/ifi/english/research/projects/ocarina

like LGC [25]: we used the same ECN signaling as used by DCTCP, but with a lower marking threshold, resulting in a smoother behavior and shorter queue length. We are now working on a multihop congestion controller based on LGC for RINA, which can benefit from the flow aggregation capability that lower DIFs provide; this enables IoT devices, depending on their application, to perform *data aggregation* [26] as well, which can significantly reduce energy consumption; this is as opposed to *packet aggregation* [27] that is performed at the physical (or a lower) layer to decrease the protocol overhead [28], but the two approaches can be employed at the same time, especially in RINA. The outline of a recursive congestion control mechanism is depicted in [29]. In addition, devising appropriate SDU Protection policies as well as multicast protocols tailored for constrained IoT devices is indeed interesting.

VI. Conclusion

In this paper, we discussed some main architectural performance and security issues of IoT networks. We showed the challenges in current IoT network stacks and why achieving a secure transport is a trade-off. Then, we discussed the recursive idea behind RINA, a promising network architecture, and explained its main modules for data transfer. Combined with embedded security at each layer, we demonstrated how transport layer security does not and should not architecturally contradict efficiency. In each layer, the EFCP module of RINA provides features hardening inside attackers. Between layers, security is ensured by SDU protection; separating these two allows any protocol translation / connection split in the path.

We presented two use cases that can benefit from our proposed solution. In both cases that an IoT device can run the RINA stack locally or not, we illustrated architectural solutions, which also facilitates performing a secure multicast; this has been a serious demand in the IoT, but without any support by current protocols. We also discussed different deployment methods for RINA and our future directions.

Acknowledgment

The work was funded by the Research Council of Norway under its "Toppforsk" programme through the "OCARINA" project. The views expressed are solely those of the authors.

References

[1] F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, "Internet of things security: A survey," *Journal of Network and Computer Applications*, vol. 88, pp. 10 – 28, 2017.

[2] O. Garcia-Morchon, S. Kumar, and M. Sethi, "Internet of Things (IoT) Security: State of the Art and Challenges," RFC 8576, Apr 2019. [Online]. Available: https://rfc-editor.org/rfc/rfc8576.txt

[3] P. I. R. Grammatikis, P. G. Sarigiannidis, and I. D. Moscholios, "Securing the internet of things: challenges, threats and solutions," *Internet of Things*, vol. 5, pp. 41–70, 2019.

[4] J. Granjal, E. Monteiro, and J. S. Silva, "Security for the internet of things: A survey of existing protocols and open research issues," *IEEE Communications Surveys Tutorials*, vol. 17, no. 3, pp. 1294–1312, thirdquarter 2015.

[5] J. Day, *Patterns in Network Architecture: A Return to Fundamentals*. Prentice Hall, 2007.

[6] P. Teymoori, M. Welzl, G. Stein, E. Grasa, R. Riggio, K. Rausch, and D. Siracuss, "Congestion control in the Recursive Internetwork Architecture (RINA)," in *IEEE International Conference on Communications (ICC), Next Generation Networking and Internet Symposium*, May 2016.

[7] M. Amadeo, C. Campolo, J. Quevedo, D. Corujo, A. Molinaro, A. Iera, R. L. Aguiar, and A. V. Vasilakos, "Information-centric networking for the internet of things: challenges and opportunities," *IEEE Network*, vol. 30, no. 2, pp. 92–100, 2016.

[8] T. Ramezanifarkhani and P. Teymoori, "Securing the internet of things with recursive internetwork architecture (rina)," in *2018 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2018, pp. 188–194.

[9] Y. Yang, L. Wu, and et al., "A survey on security and privacy issues in internet-of-things," *IEEE Internet of Things Journal*, 2017.

[10] J. Day, I. Matta, and K. Mattar, "Networking is IPC: a guiding principle to a better internet," in *Proc. ACM CoNEXT*, 2008, p. 67.

[11] J. Suarez, J. Quevedo, I. Vidal, D. Corujo, J. Garcia-Reinoso, and R. L. Aguiar, "A secure iot management architecture based on information-centric networking," *Journal of Network and Computer Applications*, vol. 63, pp. 190–204, 2016.

[12] R. Ravindran, Y. Zhang, L. A. Grieco, A. Lindgren, J. Burke, B. Ahlgren, and A. Azgin, "Design Considerations for Applying ICN to IoT," Internet Engineering Task Force, Internet-Draft draft-irtf-icnrg-icniot-03, May 2019, work in Progress.

[13] C. Fang, H. Yao, Z. Wang, W. Wu, X. Jin, and F. R. Yu, "A survey of mobile information-centric networking: Research issues and challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, 2018.

[14] J. Huang, Q. Duan, Y. Zhao, Z. Zheng, and W. Wang, "Multicast routing for multimedia communications in the internet of things," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 215–224, 2016.

[15] G. Boddapati, J. Day, I. Matta, and L. Chitkushev, "Assessing the security of a clean-slate internet architecture," in *20th IEEE International Conference on Network Protocols (ICNP)*. IEEE, 2012.

[16] E. Grasa, O. Rysavy, O. Lichtner, H. Asgari, J. Day, and L. Chitkushev, "From protecting protocols to layers: Designing, implementing and experimenting with security policies in rina," in *2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–7.

[17] S. Leon, J. Perello, and et al., "Benefits of programmable topological routing policies in rina-enabled large-scale datacenters," in *Global Communications Conference*. IEEE, 2016.

[18] J. Small, "Patterns in network security: An analysis of architectural complexity in securing recursive inter-network architecture networks," Master's thesis, Boston University Metropolitan College, 2012.

[19] "Pouzin society," 2019. [Online]. Available: http://pouzinsociety.org/

[20] R. W. Watson, "Timer-based mechanisms in reliable transport protocol connection management," *Computer Networks (1976)*, vol. 5, no. 1, pp. 47–56, 1981.

[21] M. Rizinski, J. Day, and L. Chitkushev, "Iot architecture based on rina," in *7th International Workshop on RINA, co-located with IEEE ICIN 2020*. Paris: IEEE, 2020.

[22] K. Ciko and M. Welzl, "First contact: Can switching to rina save the internet?" in *6th International Workshop on the Recursive InterNetwork Architecture (RINA 2019), co-located with IEEE ICIN 2019*. IEEE, 2019, pp. 37–42.

[23] K. Ciko, M. Welzl, and M. Marek, "Taps and rina: Do they fit together?" in *7th International Workshop on RINA 2020, co-located with IEEE ICIN 2020*. Paris: IEEE, 2020.

[24] D. A. Hayes, P. Teymoori, and M. Welzl, "Feedback in recursive congestion control," in *European Workshop on Performance Engineering*. Springer, 2016, pp. 109–125.

[25] P. Teymoori, D. Hayes, M. Welzl, and S. Gjessing, "Even lower latency, even better fairness: Logistic growth congestion control in datacenters," in *IEEE LCN*. IEEE, 2016, pp. 10–18.

[26] P. Teymoori, M. Kargahi, and N. Yazdani, "A real-time data aggregation method for fault-tolerant wireless sensor networks," in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, 2012.

[27] P. Teymoori, A. Dadlani, K. Sohraby, and K. Kim, "An optimal packet aggregation scheme in delay-constrained ieee 802.11n wlans," in *2012 8th International Conference on Wireless Communications, Networking and Mobile Computing*, 2012, pp. 1–4.

[28] P. Teymoori, N. Yazdani, S. A. Hoseini, and M. R. Effatparvar, "Analyzing delay limits of high-speed wireless ad hoc networks based on ieee 802.11n," in *2010 5th International Symposium on Telecommunications*, 2010, pp. 489–495.

[29] M. Welzl, P. Teymoori, S. Gjessing, and S. Islam, "Follow the model: How recursive networking can solve the internets congestion control problems," in *2020 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2020, pp. 518–524.