Securing the Internet of Things with Recursive InterNetwork Architecture (RINA)

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

***Abstract*—Communication technology improvements have inspired the idea of connecting almost every things to the Internet: from home appliances, medical devices, and cars, to large infrastructures. A unified and secure network of these things is almost a dream because the Internet has not had this goal from the beginning; protocols have been implemented and then secured, and then extended to new domains. This has been the cause of many vulnerabilities so far. In this paper, we take a fundamental look at the inherited architectural security issues of Internet of Things (IoT) which have raised serious security concerns due to its overwhelming number of nodes. Then, we investigate Recursive InterNetwork Architecture (RINA), a very promising network architecture, as a design solution; we demonstrate how RINA can specifically address security challenges of IoT networks, and how it mitigates their common attacks. Moreover, we will show how RINA can provide other features which are now mentioned as the future trend in IoT.**

I. Introduction

Security is a critical challenge, especially in Internet of Things (IoT) and more clearly, (from Cisco) Internet of Every Things (IoE). "Things" including electronic devices, software, sensors, and actuators are connected and communicate with each other, enabling them to be deployed in a variety of environments and domains such as home and buildings, smart infrastructure, health, and mobility, i.e. in the whole society. Being highly deployable in one hand, and the lack of security preservation in devices, with the potential risk of a large number of unsecured connected devices on the other hand, cause the concept of IoT. The number of IoT devices is predicted to reach 41 billion in 2020 with an $8.9 trillion market [1]. Communication between IoT devices through the "smart, connected products" needs a strong, secure, and scalable communication infrastructure.

While current infrastructures have difficulties in diversity of services and configurations, management of network communication for smart infrastructure becomes a bigger challenge; one of the main challenges is that existing networks and systems are generally not homogeneous and they are hardly able to communicate with each other. In addition, secured communications between systems is another significant challenge. What is needed is a well-established secure communication across different industrial domains to provide *domain synergy*. Unification plays a significant role to foster reusability and interoperability in systems, and it reduces extra costs.

Comprehensive studies such as [2], [3] on IoT security have confirmed that many vulnerabilities stem from the currently-employed network stack protocols of IoT devices and their interoperability issues with the Internet. For example, although DTLS can provide security at the transport layer, it makes difficulties in the operation of CoAP proxies; IPSec can be used at the network layer, but it hides all the transport information which affects network performance in wireless environments [4]; and provided (security) functionalities by current protocols are not sufficient [5]. Although some research work (e.g. [6], [7]) focused on presenting a secure IoT architecture, the presented architecture usually operates at higher layers regardless of what the network stack is. On the contrary, in this paper, we fundamentally looked at the security issues of lower layers, and especially the network stack and its protocols.

To address the above issues in IoT networks, in this paper, we show how Recursive InterNetwork Architecture (RINA) can be effectively employed as a secured network stack, and how much its recursive architecture can facilitate domain synergy through a unified interoperable and programmable architecture. RINA was first introduced in [8] and then, implemented and evaluated by a number of international projects. Through a number of research work (e.g. [4], [9], [10], [11], [12], [13]), it has demonstrated a very promising potential in improving network performance and security. RINA uses the idea of recursion; it defines 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) in the RINA vocabulary. The mechanisms in DIF can be customized through *policies*. Then, DIFs can be recursively reused or arranged arbitrarily; they can be stacked, chained, or put side-by-side [4]. The recursion reduces the number of required protocols, mechanisms, and security efforts at different network layers since the same program is instantiated. Since every DIF (layer) is naturally a "secured container", security is a byproduct of RINA [9].

In Section II, we discuss current security challenges of IoT; we focus on the network protocol stack and evaluate issues of common protocols employed in each layer, and we illustrate how the issues are rooted in the architectural design of the stack. In Section III, we discuss RINA, its security features, and how it can address IoT security issues through its novel architecture. In Section IV, to demonstrate effectiveness of RINA, we adopt common categories of security issues of IoT, and map RINA security features to the issues and security challenges. Then, we discuss in details how RINA, as a novel

architecture, can be deployed in IoT networks. In addition, RINA has been shown effective in addressing future trends in networks such as mobility in IoT. We also believe that RINA provides a flexible structure and makes possible to build large systems from smaller ones in multiple, different domains. As a result, different security solutions from different contexts would be easily adjusted and reused in other contexts. This facilitates composability of systems and systems of systems originally from heterogeneous environments. Finally, Section V concludes the paper.

II. IoT Security Challenges

Fig. 1 illustrates a typical network stack of IoT with common protocols in each layer; up to the MAC layer, IEEE 802.15.4 [14] is usually supported; the 6LoWPAN protocol [15] enables the transmission of IPv6 over IEEE 802.15.4; RPL [16] provides routing over 6LoWPAN; TLS [17] and DTLS [18] represent transport layer security using TCP and UDP, respectively; and CoAP [19] provides web transfer at the application layer over UDP (DTLS) with some limited congestion control features. Since we focus on the network stack of IoT devices, we do not discuss *perception* (to perceive the environment with technologies such as RFIS and GPS) which is usually categorized as a layer.

Although there has been much research work on securing IoT, there are still open issues regarding the layers, protocols, and their vulnerabilities; these issues are thoroughly discussed by [20], [3]. We argue that IoT security challenges mostly stem from the architectural design of IoT network stack. The challenges (named C_i) and their cause are as follow:

C_1 *IoT Network Stack Issues*: Despite the lessons learned from the Internet, the IoT network stack has similar security issues. Referring to Fig. 1, the 6LoWPAN protocol does not define any security mechanism, but it makes the use of IPsec available to provide security between two communication end points. However, no specific method has been adopted yet for 6LoWPAN [20]. Since the 6LoWPAN Border Router typically does not perform any authentication, IoT networks are still vulnerable [3]. Moreover, encryption at the routing layer hides all the necessary information of the upper layers; this is one of the main problems that Performance Enhancing Proxies (PEPs) [21] are facing in wireless networks because they cannot break the end-to-end congestion control loop to start a new one matching the environment properties (e.g. wired, wireless) [22]. Although there have been proposals on smart gateways (e.g. [23], [24]) to connect *things* to the Internet, the same problem still exists [20].
At the routing layer, RPL [25] is commonly used, which does not define how to protect RPL communications and operations from internal attackers, and it also lacks some key security features [20].
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 [20], 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.

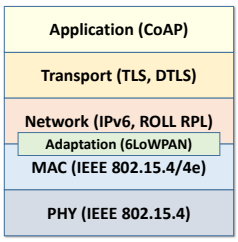

Fig. 1. A typical IoT network stack with common protocols.

C_2 *Repeated Functionality in Layers/Protocols*: As we see, almost the same (security) functionality needs to be repeated/employed/implemented at different layers several times per protocol while there is "no need to reinvent the wheel" in several layers. This repetition increases the risk of making new mistakes/vulnerabilities, and reduces the reuse degree for future extensions.

C_3 *Global, Public, and Large Address Space*: Due to the overwhelming number of *things*, IPv6 was employed. It also comes with its own challenges such as large, public addresses since each *thing* should be addressable publicly. Knowing target addresses indeed increases security challenges in IoT. Since IP addresses are globally public, adopting them is complicated due to their large size, while there is no need to expose everything to the widest scope, i.e. the Internet.

C_4 *Security and Performance Enhancement Conflict*: Performance enhancement methods such as PEPs and CoAP gateways are affected inversely by adding security features to other layers especially, transport. In other words, adding (security) functionalities to one layer might interfere proper (and mostly performance) operations at the other layers [20].

C_5 *Attack Repetition*: Current security functionalities of IoT protocols and their shortages are commonly known, and they still need extensions [3]. The overwhelming recent attacks such as DDoS in IoT devices show how historical attacks are renewed with even more power.

C_6 *Future Extensions*: There are also some other issues regarding the security of, for example, mobile devices which have not been addressed yet [26]. Even securing a protocol/layer in IoT does not mean that it is compliant to other protocols/layers or any future extensions including mobility, multicast, and QoS [27].

C_7 *Domain Synergy*: In cross-domain applications and heterogeneous environments, diversity of protocols and policies and their communication especially from the security point of view is a new challenge.

III. Recursive InterNetwork Architecture (RINA)

A. What is RINA?

RINA [10] is a novel architecture of networking; it follows a ground-breaking approach to layers in the network protocol stack that we are already familiar with. RINA was first introduced by John Day as a pattern in network architecture in [8]. To have a closer look at what RINA is, first we discuss the Internet architecture. The global network which is commonly known as "The Internet" is the network of networks connecting millions of devices together. Some approaches such as TCP/IP and the OSI model were presented to layer functionalities and provide abstractions. However, the models were deviated during development and improvement, and more importantly, the Internet was not originally designed securely; TCP/IP networks had to adopt many other protocols with redundant functionalities to be able to work. This all has shown to be complicated to manage or secure [13].

On the contrary, RINA adopts the basic foundation of networking: "Networking is Inter-Process Communication (IPC) and only IPC" [10]. It unifies networking and distributed computing: the network is a distributed application that provides IPC. Moreover, it employs a secured layer with basic IPC *mechanisms* (i.e. necessary functionalities), and through a common API, the network administrator is allowed to arrange/stack these secured layers as needed recursively. Each layer is called Distributed IPC Facilities (DIF), and is able to be programmed through *policies* on-the-fly; policies determine how mechanisms could operate.

B. RINA Structure and Mechanisms

On the contrary to the other network stacks, RINA recurses the same layer which is called DIF. The lowest layer, shim DIF, operates over any lower layer, which could be physical, or other protocols such as TCP or UDP. DIFs are usually numbered from 1 (e.g. 1-DIF, 2-DIF) as the lowest, and N-DIF refers to the current layer one focuses on.

In Fig. 2, a sample arrangement of two layers of DIFs between two end-nodes and two routers is shown. An IPC Process (IPCP) is an instance of the same code managing IPC in each layer at each node. The internal structure of every IPCP is the same, and it consists of the following mechanisms which operate at different timescales:

- Data Transfer: handles packet transmission including:
  - Delimiting: a mechanism for encoding Service Data Units (SDUs) coming from the upper layer/DIF within PDUs.
  - Error and Flow-Control Protocol (EFCP): is composed of two sub-protocols, Data Transfer Protocol (DTP) and Data Transfer Control Protocol (DTCP), which handle data transmission.
  - Relaying and Multiplexing Task (RMT): routes packets to output ports of the DIF or upwards.
  - SDU Protection: intended for encryption, compression, error-code and TTL.

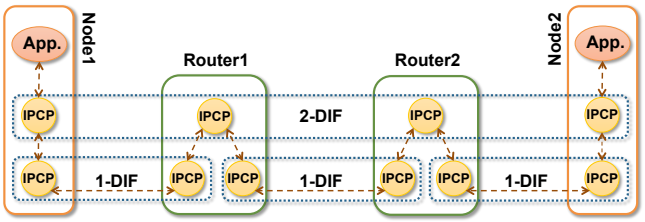

Fig. 2. A sample RINA topology with two end-nodes and two routers. Every IPCP has the same internal structure.

- Data Transfer Control: handles error, flow, and retransmission control.
- Layer Management: includes
  - Routing,
  - Common Distributed Application Protocol (CDAP): operates on configuration objects, and layer management,
  - Resource and flow allocation,
  - Locating applications,
  - Security management, access control,
  - Enrollment, and authentication.

Referring to Fig. 2, the dashed arrows show the path of data/message exchange between the two applications in nodes 1 and 2. Every IPCP decides how to process a received PDU from the upper/lower layer; it can pass it to the lower layer, send it to the other IPCP if the DIF is on the physical medium (i.e. it is a 1-DIF), routes it to another IPCP in a lower DIF if it knows where the destination is (e.g. the IPCPs in 2-DIF in the routers), or pass it upwards to the destination application (at node 2). Therefore, there is a single type of layer with programmable functions, that repeats as many times as needed by network designers. It means that all layers provide the same mechanisms: instances or communication (flows) between two or more application instances, with certain characteristics (delay, loss, in-order-delivery, etc). However, the mechanisms are programmable/customizable through policies. In general, there are only 3 types of nodes in RINA: hosts, interior[1] and border routers, and there is no need for middleboxes such as firewalls, NATs, etc. because policies can customize the internal behavior of each IPCP (or DIF), which consequently empowers nodes with any required functionality.

C. RINA Security Features

In addition to the above, RINA improves security as the byproduct of its design. As mentioned by [10], [11], [9] in detail, these features are summarized as follows:

F_1 *Secure DIFs*: "DIF is a secure container" [9]. RINA secures layers instead of protocols. Every PDU leaving N-DIF towards an (N-1)-DIF can be protected by the SDU protection module meaning that RINA protects N-DIF PDUs in their entirety as they cross the N-1 DIF boundary. Even control information (addresses, flow-ids, etc.) can be protected from the layer below. This solves

[1]Interior routers are not shown in the figure. See [11] for a complete topology.

the PEP problems with breaking secure connections by intermediate nodes [4].

F_2 *Divide and Conquer*: Through DIFs and recursion, the problem of securing a wide scope (e.g. as wide as the Internet scale) will be divided to the problem of securing smaller scopes. Compromising the protection of some DIFs does not compromise the whole network [10].

F_3 *Hidden Addresses*: In RINA, applications cannot observe addresses; each DIF has its own addressing which is hidden from other underlying DIFs. On the other hand, IP addresses are public in the Internet with no authentication.

F_4 *Communication via a Common DIF*: On the contrary to the Internet, two applications are only able to communicate if they have a DIF in common. Otherwise, they should join or create a common DIF.

F_5 *Authentication*: Every IPCP should be authenticated first before joining a DIF. This is performed before connection management through the enrollment process, and enrollment does include access control. This means that attackers have to join a DIF to be able to address IPCPs in that DIF which requires authentication first.

F_6 *Firewall*: Every router will naturally play as a firewall in RINA. Security modules in IPCPs can provide firewall functionalities.

F_7 *Programmable DIFs*: Any new functionality, which might address some security, privacy, or performance issue, can be simply developed as a *policy* and plugged into existing mechanisms. This reduces functional redundancies in protocols and the risk of causing new vulnerabilities by reducing required efforts.

F_8 *Access Control*: Authorization in RINA is performed by the Access Control module in IPCP, and uses CDAP as the signaling protocol. This mechanism determines if a requesting entity is allowed to access a given resource.

F_9 *Synchronization-Independent Port Allocation*: RINA decouples port allocation from the synchronization process happening in protocols such as TCP, which reduces the chance of intercepting a connection and makes attacks harder to mount.

F_{10} *Port-Independent Communication*: In RINA, there is no well-known ports to listen to; applications are requested for service through their application name.

F_{11} *Soft-State Connection Management*: In RINA, there is no explicit control messages for connection establishment/close. The state is deleted at the receiver after 2∗MPL (Maximum Packet Lifetime) which reduces connection management misuse [9].

F_{12} *Connection Management Independent Authentication*: Authentication is decoupled from and performed before connection management in RINA. This means that just insiders (authenticated IPCPs in a DIF) can attack.

F_{13} *Insiders Resistance*: RINA uses a wider range of control field values (e.g. connection/QoS id). Given that an attacker can somehow compromise authentication or without the support of cryptography, 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 [9].

F_{14} *QoS*: Every connection in RINA is established after the source represents its QoS requirements which include maximum requested bandwidth [28]. Deviating from those, e.g. in DoS attacks by congesting the network, can result in dropping its packets at the first routing node, which is some form of DoS prevention.

F_{15} *Variable Address Space*: Every DIF has its own address space, which could be smaller or larger, depending on the number of nodes in that DIF. This saves more space in the packet header. Hence, not only the addresses are not clear for attackers, but also the address length is not known which results in another obstacle in length attacks.

F_{16} *Mobility*: Mobility management in RINA is smoothly performed since every IPCP at every DIF can seamlessly join/leave DIFs without losing its name in its own DIF. It just needs some local routing updates at lower DIFs, without any side-effects on security [10]. In addition to mobility, RINA can also improve multi-homing [29].

F_{17} *Resiliency*: In each DIF, (multi-path) routing is performed independently and transparently to the other DIFs. This means that each DIF can provide resiliency services as well to the upper DIFs. In addition, this property provides "transport over heterogeneous networks" [30].

F_{18} *Performance Improvements*: In addition to the above security features, RINA has some other important features which are all very appealing for IoT networks [31]. For example, through some research work[2] and international projects[3], it has been shown that RINA can effectively improve the network performance in terms of throughput and delay without compromising security [4].

F_{19} *Complexity Reduction*: Considering the number of protocols, required flows, and especially required distinct mechanisms, RINA networks can satisfy security requirements with less complexity than in the current Internet. Moreover, the number of active instances of networking mechanisms is reasonably less complex in RINA with a secured link layer [13]. Rina also reduces the size of routing tables [12], [32], [33].

F_{20} *Arbitrary Arrangement*: DIFs can be arranged/stacked arbitrarily to provide different operations such as PEPs, multipath routing, and in-network resource sharing without compromising security [4].

IV. Employing RINA for IoT Security

A. How RINA Addresses Security Challenges

RINA approaches problems in a divide-and-conquer manner; it defines different scopes (DIFs) with their own security. Every N-DIF (including its IPCPs) is responsible for its security; insider IPCPs are all trusted/pre-authenticated. DIFs

[2]A complete list of publications on RINA can be found in http://www.pouzinsociety.org/research/publications

[3]See http://www.pouzinsociety.org/research/projects

can be stacked arbitrarily without the need for developing any new layer/protocol. When a DIF is secured, the same code is reused for upper layers. This mimics tunneling in the Internet.

A classification of IoT device attacks includes physical, network, software, and encryption attacks [2]. Physical attacks can be performed by short-distance attackers and a part of the countermeasure is to verify the device authentication [2]. Although RINA is vulnerable to insider attacks, each device in the environment has to authenticate itself before communication which can prevent this kind of attacks. In addition, devices should employ an error detection system, and all of their information has to be encrypted to maintain data integrity and confidentiality, which is possible through DIF programmability. For network attacks, authentication mechanisms and point-to-point encryption are proposed to ensure privacy of data and rooting security. Again, RINA authentication mechanisms in addition to the possibility of utilizing other security mechanisms such as encryption via SDU protection is suitable to defend against these attacks. Since RINA nodes (IPCPs) also play the role of firewall, they can prevent illegal data access or harming the system against application and software-based attacks despite their vulnerabilities; this could be thought as a complementary defense in the presence of other tools such as anti-viruses. However, the last category which is called encryption attacks can be prevented as before by using other existing mechanisms in RINA.

In Table I, we summarize the most security challenges of IoT networks, and how RINA can address those using its features. We focused on a number of common network and application attacks, security properties, and the IoT architectural challenges discussed in Section II. The attacks category is taken from [5] which discusses in detail how they affect IoT networks.

Denial of Service (DoS) and Distributed DoS attacks are historically considered as one of the major security threats and among the hardest security challenges. Although there are lots of proposed defense mechanisms against them such as packet filtering or intrusion detection systems, they are making the headlines frequently and have become the hugest cyberattacks. In addition, they are improved and extended several times in different platforms, e.g. in case of Mirai attack [34].

Referring to the table, DoS attacks can be prevented/mitigated by F_5, F_{10}, F_{11}, and F_{14}. This means that the attacker cannot be an outsider (F_5); he/she should join the DIF first or create a new one. Moreover, the preceding step in these attacks is flooding, but there is no listening port to be the target of such attacks (F_{10}). In addition, since connections in RINA do not need to wait for explicit control messages to terminate (F_{11}), flooding attacks and their impacts are significantly mitigated. Even for an insider, there are mandatory QoS requirements that the flow should obey (F_{14}), and any deviation from that by the sender can result in dropping packets at routers.

To defend against spoofing attacks such as IP (address) spoofing that exploits valid and authorized IP addresses, RINA hides internal DIF addresses (by F_3) and decouples port allocation from the synchronization (by F_9), and also F_5 makes sure that no DIF outsider can attack.

TABLE I
RINA FEATURES AGAINST IOT SECURITY CHALLENGES

Challenges			**RINA Features**
Attacks	**Network**	Denial-of-Service	F_5, F_{10}, F_{11}, F_{14}
		Spoofing	F_3, F_5
		Sinkhole	F_1, F_2, F_6
		Wormhole	F_1, F_2, F_6
		Man in the Middle	F_1, F_5, F_9, F_{13}
		Routing Information	F_1, F_3, F_5, F_6
		Sybil	F_1, F_5
		Unauthorized Access	F_5, F_8, F_{12}
	Application	Phishing	F_1, F_5, F_8
		Malicious Virus/Worm	F_5, F_6
		Malicious Scripts	F_7
Privacy			F_1, F_4, F_7
Authentication and Nonrepudiation			F_5
Access Control			F_8
Integrity and Confidentiality			F_1
Architectural		C_1: Network Stack	F_1, F_2, F_{17}, F_{19}
		C_2: Rep. Functionality	F_1, F_7, F_{20}
		C_3: Address Space	F_2, F_3, F_{15}
		C_4: Sec.&Perf. Conflict	F_1, F_2, F_{20}
		C_5: Repeating Attacks	F_1, F_5, F_{13}
		C_6: Future Trend	F_7, F_{14}, F_{16}, F_{17}, F_{18}
		C_7: Domain Synergy	F_1, F_2, F_4, F_{17}, F_{20}

In sinkhole attacks, a compromised device tempts the others to use that them in a data routing process. In addition to secure routing protocols (provided by F_1, F_2) useful to prevent these attacks, the capability of having firewall functionality in routers, i.e. F_6 is helpful. This feature (F_6) can similarly prevent wormhole attacks as well. In addition, since performing some attacks such as DoS is the prerequisite of the others to compromise a device, prevention of DoS attacks can be used to prevent sinkhole attacks. However, to keep the table simple, we have mentioned the least features that can directly satisfy a security requirement, solve a challenge, or prevent an attack. For the rest of the network attacks in Table I, it is shown how security features of RINA can prevent/mitigate the attacks, which is straightforward, and hence, we skip discussing them.

For the application attacks such as phishing, authentication (F_5) and authorization (F_8) can be used for mitigation. The prevention of the other application attacks are also shown in the table. Preserving the security requirements shown in the table will prevent some attacks. For example, confidentiality preservation can mitigate the impact of malicious scripts.

As discussed by a lot of work such as [5], [26], preserving privacy is another important issue in IoT. We see that RINA provides a simple way of addressing privacy; for example, RINA can use a wider range of control field values for privacy tags. Moreover, due to the ability of policy enforcement in RINA, i.e. F_7, privacy-preserving access control can be applied. In addition, a common DIF with its invisible addresses (F_4) can be used to preserve anonymity and privacy policies. As we also see in the table, RINA can address confidentiality and integrity through RINA SDU Protection module [13] (feature F_1) as well as other security challenges in IoT networks through a unified, recursive architecture. Authentica-

tion, access control, and integrity and confidentiality are also addressed by F_5, F_8, and F_1, respectively. Moreover, for these requirements, RINA has specific security modules [13].

The table also discusses how RINA can address the architectural security challenges of IoT we introduced in Section II. For example, the secure DIFs (F_1) and divide and conquer (F_2) features in RINA can solve the problem of non-considered security in stack layers. For instance, non-supporting authentication in 6LoWPAN is solved by the authentication support in RINA security modules; the transport multicast support and proxy compatibility are simply addressed by the F_{17} and F_1, respectively because each DIF has its own security/routing, and different arrangements of DIFs, as shown in [4], and their reduced complexity (F_{19}) can simply provide the above requirements.

C_2 is naturally solved by RINA with DIF recursiveness (we can arbitrarily arrange/stack secure DIFs) and programmability (DIFs can be customized via policies), and there is no need to develop new protocols as the same code (IPCP) is instantiated recursively[4]. C_3 is handled by creating DIFs to divide the widest scope into smaller, simpler scopes (DIFs) by F_2 with their own hidden and variable-length address space (F_3, F_{15}). C_4 is simply solved by the property that in RINA, each DIF is inherently a "secured container" (F_1), and DIFs can be arranged/stacked without compromising the security of each other (F_2, F_{20}). Therefore, proxy operations do not interfere the security of other layers. C_5 is usually prevented by forcing users/IPCPs to enroll themselves first in secure DIFs (F_1, F_5), and it is also more difficult for DIF insiders to perform attacks (F_{13}). Any future need, C_6, can be addressed through RINA's programmability (F_7) which is inherent in the current RINA architecture, and the other features such as QoS (F_{14}), mobility (F_{16}), and resiliency (F_{17}) without compromising security. Finally, domain synergy (C_7) can be simply handled by the recursion property of RINA (F_2, F_{20}) and secure, common DIFs (F_1, F_4) without any side effects on the underlying security/performance, regardless of the beneath network environment (F_{17}).

Domain synergy and how to have a reference architecture is an important challenge in IoT. For example, we are working on such a reference architecture in the SCOTT project[5] to foster security, reusability, scalability, and interoperability. The objectives are to leverage the future IoT design middleware mechanisms and the supporting tools needed between different industrial domains with different requirements. Based on the unification provided by RINA, a reference architecture can be conducted similarly with higher reusability and manageable policies across different domains. In addition, in the huge and cumbersome synergies between domains with different reference architectures, based on some work such as [13], RINA DIFs promote reducing the scope of networks significantly. However, there are still some challenges such as traditional authentication and authorization methods that may not be applicable to the IoT because of heterogeneity and complexity of objects. Moreover, due to hidden addresses for applications in RINA, end-to-end authentication and authorization may encounter new issues. We aim to develop a lightweight and compatible structure of attribute-based access control policies [35] for DIFs to overcome these issues. To practically analyze how RINA is effective to prevent some existing security challenges, we are focusing on some attractive IoT devices by attackers such as CCTV cameras that are widely vulnerable to simple hacks, and we are developing an approach utilizing programmable DIFs to defeat applicable attacks to these devices. We are also developing metrics for measuring the security level of communication to evaluate the RINA architecture in preventing attacks.

B. RINA Deployment Considerations

One of the main issues in the design of the IoT network stack is *interoperability*, i.e. how to guarantee that IoT devices can communicate with existing Internet applications and follow Internet standards [20]. This has made them adopt many existing protocols and apparently, inherit their vulnerabilities and design issues.

Adopting RINA, as a new protocol stack, does imply interoperability considerations which are now under investigation and implementation by some projects such as OCARINA[6] that we are working on. As proposed by OCARINA and also [36], RINA can be deployed as an overlay/underlay/alongside other networks including the Internet; as an overlay, RINA can operate on all PHY, Link, IP, and TCP/UDP layers through its shim DIFs; as an underlay, it can seamlessly transmit, for example, TCP/IP traffic; and alongside other networks, through simple proxy IPCPs, it has been shown how RINA inter-operates with other network stacks. It has also been investigated how RINA can operate on tiny, limited devices such as wireless sensors in the RINAiSense project[7].

V. Conclusions and Further Work

The trend towards connecting *everything* to each other and to the Internet has raised serious security concerns. In this paper, we discussed some main architectural security issues of IoT networks: security and privacy issues, security limitations of the current network stack and its employed protocols, domain synergy, future directions, and expectations from IoT networks.

RINA shows to be a promising network architecture that has shown significant improvements in many security and performance aspects. We briefly discussed RINA modules and security features. By investigating RINA for current IoT attacks, security requirements and challenges in IoT networks, we showed that RINA has architectural solutions for each problem. In addition, it is programmable through policies which help extend its mechanisms. We believe that the *recursiveness* of RINA and the *natural security* of each recursion enables us to build arbitrarily-large secure IoT.

[4]All IPCPs in Fig. 2 are instantiated from the same code.

[5]http://its-wiki.no/wiki/SCOTT:Home

[6]See http://www.mn.uio.no/ifi/english/research/projects/ocarina

[7]See https://distrinet.cs.kuleuven.be/research/projects/RINAiSense

Despite its maturity level, the development of security protocols for the Internet is still in progress. Likewise, in RINA some aspects such as key exchange/management have been recently implemented, and the level of trust in management data, and integrity-protecting routing are also under active development. We consider further evaluation of these features as our future work.

ACKNOWLEDGMENT

The research leading to these results has received funding from the SCOTT - Secure Connected Trustable Things. SCOTT (www.scottproject.eu) has received funding from the Electronic Component Systems for European Leadership Joint Undertaking under grant agreement No 737422. This Joint Undertaking receives support from the European Unions Horizon 2020 research and innovation programme and Austria, Spain, Finland, Ireland, Sweden, Germany, Poland, Portugal, Netherlands, Belgium, Norway. Peyman Teymoori 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] "Why the internet of things is called internet of things: Definition, history, disambiguation," 2014, *https://iot-analytics.com/internet- of-things-definition/* .

[2] I. Andrea, C. Chrysostomou, and G. Hadjichristofi, "Internet of things: Security vulnerabilities and challenges," in *Computers and Communication (ISCC), 2015 IEEE Symposium on*. IEEE, 2015, pp. 180–187.

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

[4] 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.

[5] J. Lin, W. Yu, and et al., "A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet of Things Journal*, 2017, doi:10.1109/JIOT.2017.2683200.

[6] 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. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804516000370

[7] I. D. Addo, S. I. Ahamed, S. S. Yau, and A. Buduru, "A reference architecture for improving security and privacy in internet of things applications," in *2014 IEEE International Conference on Mobile Services*, June 2014, pp. 108–115.

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

[9] 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.

[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] 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.

[12] 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.

[13] 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.

[14] "Ieee standard for local and metropolitan area networks–part 15.4: Low-rate wireless personal area networks (lr-wpans)," *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)*, pp. 1–314, Sept 2011.

[15] G. Montenegro, C. Schumacher, and N. Kushalnagar, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals," RFC 4919, Aug 2007. [Online]. Available: https://rfc-editor.org/rfc/rfc4919.txt

[16] P. Thubert and J. Hui, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks," RFC 6282, Sep 2011. [Online]. Available: https://rfc-editor.org/rfc/rfc6282.txt

[17] E. Rescorla and T. Dierks, "The Transport Layer Security (TLS) Protocol Version 1.2," RFC 5246, Aug 2008. [Online]. Available: https://rfc-editor.org/rfc/rfc5246.txt

[18] E. Rescorla and N. Modadugu, "Datagram Transport Layer Security Version 1.2," RFC 6347, Jan 2012. [Online]. Available: https://rfc-editor.org/rfc/rfc6347.txt

[19] Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)," RFC 7252, Jun 2014. [Online]. Available: https://rfc-editor.org/rfc/rfc7252.txt

[20] 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, 2015.

[21] C. Caini, R. Firrincieli, and D. Lacamera, "PEPsal: a Performance Enhancing Proxy for TCP satellite connections," *IEEE Aerospace and Electronic Systems Magazine*, vol. 22, no. 8, pp. 7–16, 2007.

[22] T. T. Thai, D. M. L. Pacheco, E. Lochin, and F. Arnal, "SatERN: a PEP-less solution for satellite communications," in *Proc. IEEE ICC*, 2011.

[23] D. Bimschas, H. Hellbrück, and et al., "Middleware for smart gateways connecting sensornets to the internet," in *Proceedings of the 5th International Workshop on Middleware Tools, Services and Run-Time Support for Sensor Networks*. ACM, 2010, pp. 8–14.

[24] B. Kang, D. Kim, and H. Choo, "Internet of everything: A large-scale autonomic iot gateway," *IEEE Transactions on Multi-Scale Computing Systems*, vol. PP, no. 99, pp. 1–1, 2017.

[25] R. Alexander, A. Brandt, J. Vasseur, J. Hui, K. Pister, P. Thubert, P. Levis, R. Struik, R. Kelsey, and T. Winter, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," RFC 6550, Mar 2012. [Online]. Available: https://rfc-editor.org/rfc/rfc6550.txt

[26] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in internet of things: The road ahead," *Computer Networks*, vol. 76, pp. 146–164, 2015.

[27] 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. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804517301455

[28] S. L. Gaixas, J. Perelló, and et al., "Assuring qos guarantees for heterogeneous services in rina networks with δq," in *Cloud Computing Technology and Science , 2016 IEEE International Conference on*. IEEE, 2016.

[29] V. Ishakian, J. Akinwumi, F. Esposito, and I. Matta, "On supporting mobility and multihoming in recursive internet architectures," *Computer Communications*, vol. 35, no. 13, pp. 1561–1573, 2012.

[30] E. Trouva, E. Grasa, and et al., "Transport over heterogeneous networks using the rina architecture," in *Proceedings of the 9th IFIP TC 6 International Conference on Wired/Wireless Internet Communications*, 2011.

[31] E. Trouva, E. Grasa, J. Day, I. Matta, L. T. Chitkushev, P. Phelan, M. P. De Leon, and S. Bunch, "Is the internet an unfinished demo? meet rina!" in *TERENA Networking Conference*, 2010, pp. 1–12.

[32] F. Hrizi and A. Laouiti, "Hierarchical small world overlay for efficient forwarding in volunteer clouds," in *IEEE 31st International Conference on Advanced Information Networking and Applications*. IEEE, 2017.

[33] F. Hrizi, A. Laouiti, and H. Chaouchi, "Sfr: Scalable forwarding with rina for distributed clouds," in *Network of the Future (NOF), 2015 6th International Conference on the*. IEEE, 2015, pp. 1–6.

[34] "New Mirai variant hits target with 54-hour DDoS," 2016, *https://www.infosecurity-magazine.com/news/new-mirai-variant-hits-target-with/*.

[35] V. C. Hu, D. Ferraiolo, R. Kuhn, A. R. Friedman, A. J. Lang, M. M. Cogdell, A. Schnitzer, K. Sandlin, R. Miller, K. Scarfone *et al.*, "Guide to attribute based access control (abac) definition and considerations (draft)," *NIST special publication*, vol. 800, no. 162, 2013.

[36] E. Grasa, E. Trouva, S. Bunch, P. DeWolf, and J. Day, "Developing a rina prototype over udp/ip using tinos," in *Proceedings of the 7th International Conference on Future Internet Technologies*, 2012.