Identification of Threats and Security Risk Assessments for Recursive Internet Architecture

Hamid Asgari, *Senior Member, IEEE*, Sarah Haines, and Ondrej Rysavy

***Abstract*—There are several types of attacks on communication networks such as disrupting or blocking communication, intercepting, injecting fabricated packets, accessing and modifying the information. Here, for the first time the SecRAM, a recent security risk assessment methodology, is proposed to be systematically applied in a different context, i.e., to the network systems, specifically to an emerging network architecture called recursive internetwork architecture (RINA). The security risk assessment is performed to: identify run-time threats; assess the impact and likelihood of occurrence of attacks relevant to the threats; evaluate the RINA design principles; and validate the built-in security enablers and the mitigation actions that are devised to combat such attacks. Resulting from this assessment, specific measures are proposed to further improve cyber resiliency of the RINA, in securing its layers and components. The enhancement prevails through the utilization of multilayered security controls or the increase in their strength. We show how programmable security controls can assist in tackling network attacks. For proof of concept, we demonstrate formal analysis of some of the security properties of RINA using ProVerif tool and RINA Simulator. We apply the tool to create a formal model of a network and mitigate the selected attacks. The results of this analysis are provided.**

***Index Terms*—Network, recursive internetwork architecture (RINA), risk assessment, security, threat.**

I. Introduction

SECURITY threats are associated to vulnerabilities of systems that can be exploited by attackers. Any intrusion or attacks on the network vulnerabilities, computer or information systems may have undesirable consequences. Existing network security approaches do not provide tight integration of network functions and security controls to properly countermeasure the attacks. Therefore, new innovative ways of protecting networks against attacks are required, i.e., by embedding the security functions in the fabric of network systems to address the above shortcoming. In new network architecture designs, programmability, the ability to dynamically configure, control, and combine the security functions, are regarded as the key features to create resilient network systems for preventing or reducing the impact of attacks. The recursive internetwork architecture (RINA) is an emerging network architecture that recognizes the need for building security enablers into the architecture and is aiming to provide the stated design features.

Methodologically, sound assessments are crucial for understanding a system in fulfilling the needs, realizing its behavior, and identifying the possible implications. In this paper, we propose the use of a structured methodological approach for identifying threats and assessing the associated risks in the network systems in general. Undertaking security risk assessment in a structured way means making more accurate design decisions about the resilience of a system's services and functions and avoiding costly security incidents during system's operation. Here, we specifically establish the context and set out the scope for the security analysis of RINA network architecture, assessing the risk levels, validating the necessary security enablers/controls that have been built into the architecture, and recommending new enhancements. For validation purposes, the defined security enablers/controls are checked against the stakeholders' security requirements and network security needs in order to meet them. Embedding security controls in the architecture for combating run-time threats is the key step toward the security-by-design concept enabling cyber resiliency and avoiding incremental updates and plug-ins. Cyber resiliency enablement allows the network systems to be resilient against persistent, stealthy attacks targeted at cyber assets [1]. The way we apply this risk assessment methodology to a network system can inspire its use in other emerging network architecture settings.

The three main security requirements/criteria specified for consideration in information systems are: to prevent unauthorized information disclosure [confidentiality (C)] and improper malicious modifications of information [integrity (I)], while ensuring access for authorized entities [availability (A)]. There are several types of attacks on network communications including: disrupting or blocking communication, intercepting, injecting fabricated packets, accessing and modifying the storage, tables or packets.

The RINA is built on a fundamental principle that networking is based on distributed interprocess communication (IPC) that repeats over different scopes, with the aim of providing IPC services to applications in a more efficient way [2]. To this end, the main attacks to the network will focus on disrupting this communication service. In this paper, by considering the above requirements, we perform the security risk assessment to identify run-time threats to the RINA network, evaluate the RINA's design foundation in addressing the security threats, and verify/enhance the specific security controls devised to satisfy the above requirements in order to mitigate the attacks.

The remainder of this paper is structured as follows. After this brief introduction in Section I, Section II provides the related work. Section III describes the risk assessment methodology

Manuscript received November 7, 2016; revised May 15, 2017 and July 28, 2017; accepted October 13, 2017. This work was supported in part by the European Commission, FP7 Collaborative PRISTINE Project, 619305. *(Corresponding author: Hamid Asgari.)*

H. Asgari and S. Haines are with the Thales UK Research, Technology & Innovation, Reading RG2 6GF, U.K. (e-mail: Hamid.Asgari@uk.thalesgroup.com; Sarah. Haines@uk.thalesgroup.com).

O. Rysavy is with the Faculty of Information Technology, Department of Information Systems, Brno University of Technology, Brno, 601 90, Czech Republic (e-mail: rysavy@fit.vutbr.cz).

Digital Object Identifier 10.1109/JSYST.2017.2765178

used for security analysis. Section IV briefly explains the RINA architecture as the context for this security analysis, the scope of the risk assessment study, and the network assets. Section V specifies the threat scenarios relevant to RINA. The security risk assessment is described in Section VI in terms of the impact, likelihood, and associated risk levels of identified threats. Section VII proposes the security controls to put in place to mitigate the threats with high and medium risk levels. A prototype has been developed to exploit RINA's programmability including its security functions. Section VIII discusses this and threat monitoring aspects. Formal analysis of RINA's security controls is provided in Section IX. Section X concludes the paper and discusses the plan on further work.

II. Related Work

A significant body of works exists in the literature on risk management. Among these works, there are established security risk assessment standards, frameworks, methodology, and guides (e.g., ISO 27005 [3]), NIST SP800-30 [4], MITRE [5], ENISA [6] that are used to aid formal risk analysis procedures in various contexts.

Among them ISO/IEC 27005 is a well-known information security risk management standard used by both commercial and government sectors. It does specify a continual process consisting of a structured sequence of activities, some of which are iterative in establishing the risk management context, assessing quantitatively or qualitatively relevant information risks, treating the risks appropriately, and monitoring and reviewing the risks. It does not specify or recommend any specific risk management method.

ISO/IEC 31010 [7] is a risk management standard that specifies 31 risk assessment techniques. There are also methods/tools developed for assessing risks such as CRAMM, MEHARI, OCTAVE, etc. Some information about the above methods is given in [8].

The AURUM [9] presents a methodology for supporting the risk management process defined by NIST SP800-30 in assessing risks in IT systems. The Single European SKY ATM Research (SESAR) SWP16.2 defined a qualitative methodology, namely security risk assessment methodology (SecRAM) [10]. SecRAM is intended for air traffic management (ATM) contexts, e.g., [11]. Another example of its application is given in [12] by building a relevant threat scenario and designing a risk treatment for a cloud-based ATM environment. There are quantitative risk assessment methodologies [13] that are objective and set out to define, measure, predict, and provide a confidence level of likelihood and occurrence of threat impacts.

Risk assessment methodologies have also been applied to cloud computing that builds heavily on capabilities through some core technologies including software services, hardware platforms, infrastructures, and networks. In addition to the stated three main security requirements, there are also a few new security requirements unique to cloud computing, introduced in the literature, including multiparty trust and mutual auditability. A quantitative impact and risk assessment approach for cloud security (QUIRC) is also proposed in [14]. It has also been discussed in [15] that security risk assessments of complex systems such as smart grid and metering networks is a challenge due to the very distributed nature of these systems. This challenge has been recognized by Smart Grid Working Groups to develop a risk assessment toolkit. It is out of scope of this paper to evaluate how secure RINA is in comparison with other network architecture models. However, in [16] authors explored the security properties of RINA, analyzing the principles that make RINA inherently more secure than TCP-/IP-based networks. It is discussed that RINA's approach to securing layers instead of protocols increases the security of networks while reducing the complexity. It is also discussed in [17] that RINA architecture makes transport level attacks (e.g., security attacks faced by TCP connections) much harder to mount. In [18], Small *et al.* explore the security of a distributed IPC facility DIF (DIF as a layer in RINA) and examine the threat model of the RINA API and its application protocol to show that a DIF is a securable container.

SecRAM is the most recent well-defined risk assessment methodology based on the ISO 27005 international standard and is compatible with existing methodologies. It facilitates integration of security from the beginning of the system development life cycle including the architectural phase. Therefore, we use the SecRAM methodology's main principles in our risk assessment study and apply it to a new context, a network system (i.e., RINA), considering all its assets and their values to the network architecture. The work presented here, we believe, is the first to systematically provide a complete description of the processes required for identifying the security threats and quantifying the risks, validating the need for built-in security functions, and proposing additional measures for reducing security risks to acceptable levels for improving cyber resiliency of a network system (i.e., RINA). These measures constitute the network system's self-protection, so that it is crucial to establish that the overall system design is resilient and reliable for large-scale use. This can also pave the way for applying it to other emerging network architecture settings.

III. Risk Assessment Methodology

The evaluation of the threats proposed here will loosely follow the SecRAM methodology. SecRAM is the ISO 27005 based risk assessment methodology developed by the SESAR program [10]. SecRAM was intended for another context (i.e., ATM) utilizing a rich set of heterogeneous systems and networks and as such it is too heavyweight to apply it in full to a network system. We, therefore, tailor the SecRAM methodology to apply it specifically to RINA. This tailoring is mainly related to the impact assessment of the identified potential attacks, which are explained in Section VI.

This methodology requires establishing the context for defining the boundaries of what one wants to analyze, sets out the scope of the security analysis, and specifies the criteria that will be used to assess the risk, in order to provide consistent and defensible results.

The security risk assessment process adheres to the following steps.

1) Establish the context and an accurate scope: describe the system that is the target of the study (RINA architecture here), its boundaries, and dependencies on other systems.
2) Identify the assets: Identify the elements that have value for the achievement of stakeholders' objectives.
3) Identify the threats and threat scenarios: Identify possible (or credible) threat sources and related scenarios to specify routes that an attacker may use to access an asset.
4) Evaluate the impact of attacks: Assess the harm resulting from an attack by taking into account the value of each asset in terms of the C, I, and A pertinence of the threat.

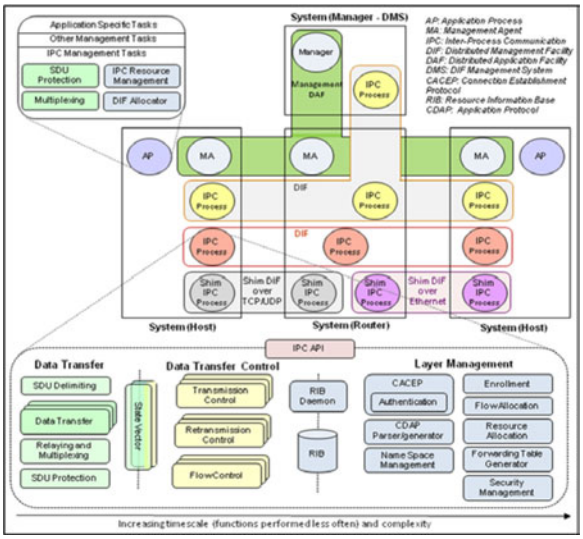


Fig. 1. RINA architecture reference model and its components.

5) Evaluate the likelihood (probability) of each threat scenario: Estimate the chance that the threat occurs and that the scenario sequence is completed successfully.
6) Assess the security risk: Evaluate the risk level associated to each combination of threat and threat scenario based on their likelihood and impact on the assets. Risk is defined as a function of the likelihood of a given threat source exercising an action on a potential vulnerability, and the resulting impact of that event on the network system.
7) Verify the risk level against the security objective: Evaluate and verify the evaluated risk level against the defined security objectives as a measurable statement of intent relating to the protection of a primary asset (PA). Security objectives correspond to the level of risk that a PA is prepared to accept on the C, I, or A criterion, before any action is deemed necessary to reduce it.
8) Risk treatment: Define the action to take, which can be to accept or tolerate the risk, reduce the risk, avoid the risk by withdrawing the activity at the root of it or transfer the risk to another party to manage it. If the action is to reduce the risk, define a set of security controls and the associated functions to reduce the risk to an acceptable level (i.e., within the risk appetite, see Table IX).
9) Security controls: Implement and put in place the security controls identified during the risk treatment step.

The prerequisite for performing a risk assessment is to clearly define the system under threat, which is represented in this study by the RINA reference model. We now apply the above process to the RINA architecture.

IV. Context, Scope, and Assets

A. Context—RINA Architecture

In this paper, we provide the security analysis focusing on RINA architecture and its components. The reference model of the RINA architecture and its components are shown in Fig. 1. The main components of the RINA architecture are the application processes (APs), distributed application facility (DAF), IPC process (IPCP), distributed IPC facilities (DIF), and the associated protocols. In this section, we briefly introduce these functional components and the protocols. Further information on the detailed functionality of these components is given in PRISTINE project deliverables [19].

An AP is the instantiation of a program executing in a processing system intended to accomplish some purpose. An AP contains one or more tasks or application entities, as well as functions for managing the resources (processor, storage, and IPCP) allocated to this AP. A DAF is a collection of two or more cooperating APs in one or more processing systems, which exchange information using IPC processes and maintain the shared state of the DAF. APs can ensure the C and I of data they pass to DIF and to the remote APs.

RINA is structured around a single type of layer called DIF. A RINA network consists of repeated DIFs. A DIF is a DAF that does IPC with the task of managing the resources in its domain. A DIF is a distributed application that performs a coordinated set of policy-managed mechanisms to provide IPC services. A DIF as a layer recurs and repeats as many times as is necessary to effectively cover the range required for the operation of the network (e.g., one or more network domains). Every DIF simply implements the same functions and uses the same protocols, but is configured with different policies to fulfill the particular requirements of the layer.

The attack surface of a layer increases with its size. The scope of IP layer across the Internet is huge, thus, making IP nodes more prone to attacks. On the contrary, the scopes of DIFs are defined by network providers enabling them to reduce the attack surface and limit the activities of adversaries.

A DIF itself is a collection of IPCPs. DIFs (i.e., layers in RINA) and their associated IPCPs are the building blocks in RINA. IPCPs join a DIF by communicating with an IPCP that is already a member of the DIF. The joining IPCP and the existing IPCP DIF member must have an underlying DIF in common, called the N-1 DIF relative to the current DIF. The joining IPCP requests the N-1 DIF to create a flow to the DIF member by issuing an Allocate_Request. The Allocate_Request, Deallocate_Request, Send, and Receive are API calls. The IPCP flow allocator in the N-1 DIF can decide whether to accept a flow request or not. If the request is accepted, the flow allocator of the N-1 IPCP attempts to create the requested flow. The flow can be destroyed by the IPCP when it is no longer needed by issuing a Deallocate_Request to the N-1 DIF. Once this flow has been established, the joining IPCP establishes an application connection to the DIF member. As part of establishing this connection, the joining IPCP must authenticate to the DIF member. If the new IPCP member is allowed to join the DIF, it is then initialized with the current information in the DIF, e.g., addressing, policies, keys, etc. Once an IPCP has successfully enrolled in a DIF, it can offer an IPC service to the layer above via the IPC API.

IPCPs perform three sets of actions: transfer of data units, data control, and IPCP management.

The data units, passed across the DIF (N) interface to be transferred to the destination AP, are called service data units (SDU). An SDU may be fragmented or combined with other SDUs for sending as one or more protocol data units (PDU). A PDU is the string of octets exchanged among the protocol machines (PM) and contains two parts: the protocol control information, which is interpreted by the PMs at the DIF, and the user-data, which is passed to its user (i.e., AP).

Data transfer is performed using the error and flow control protocol (EFCP). EFCP consists of an instance of data transfer protocol (DTP) and optionally data transfer control protocol for each data flow. When transferring data, the SDU delimiting module performs fragmentation or concatenation on the data,

depending on the flow configuration. The SDU (N) protection module applies encryption and data correction techniques for C or I protection, according to configured policies. The relaying and multiplexing task (RMT) multiplexes outgoing SDUs onto an outgoing port. It also relays incoming PDUs within the current IPCP or to an outgoing port. The IPCP management process includes support for an IPCP joining a DIF (enrollment), resource and flow allocations, and routing.

Common application connection establishment protocol (CACEP) performs the connection establishment between APs while Common Distributed Application Protocol (CDAP) or another application protocol provides the data transfer phase. CACEP is initiated by sending a CONNECT message. The connection is terminated by sending a RELEASE message.

The resource information base (RIB) is the logical representation of all information held by the IPCP or AP. Each member of the DIF/DAF (i.e., IPCP/AP) maintains a RIB as the storage. RIB should be access controlled to protect its objects. RIB Daemon is the broker for RIB that operates on objects in the RIB. The full description of reference model and RINA functional components are given in D2.2 report [19].

In RINA, network management is performed by a DAF-based DIF management system (DMS). The DMS follows a manager-agent model where a Manager process uses management agents (MA) in each network's system in order to monitor those systems and update their configuration.

The key management functions in RINA are assumed between two entities: the central key manager (KM) and the local key agent (KA). The central KM is responsible for distributing keys in a network domain and resides in the management system and may be either part of, or sit alongside, the DMS manager. The central KM can handle the entire lifecycle of key material. It also acts as a key broker that generates and stores all keys for protecting application data (i.e., SDU protection) and other key material and distributes them to a local KA on a system on request. The local KA resides on each network's system and may be either part of, or sit alongside, the MA. The same model is used for access control, i.e., access control manager (ACM) and access control agent (AcA). The description of above components and their interactions are given in D4.2 and D4.3 [19].

The RINA reference model has already provisioned for a number of security controls and functions that are a natural part of the model including authentication and SDU protection components built into the IPCPs. Authentication in an AP is not part of "IPC Management Tasks" but it is part of the application connection, which is in "Other Management Tasks," both shown in Fig. 1.

B. Scope

Establishing the context means defining the bounds of what you want to analyze. Design time identification of vulnerabilities in the specification of RINA protocols and APIs, and mitigation of these, are out of the scope of this paper. We only consider run-time attacks in order to make provision for built-in countermeasures.

We consider a simple single-domain RINA network with a DAF, an N-level DIF and an (N-1)-level DIF and the attacks originating from each of these three layers as described below (see also Fig. 2).

- An AP in the DAF relies on the N-level DIF to transport data. The AP cannot see the internals of the underlying N-level DIF and has no visibility of the N-1-level DIF; a malicious AP can only attack the N-level DIF through the IPC API. However, it is able to establish a CDAP connection to another AP in the DAF using CACEP and then send application data using CDAP.
- An IPCP can only access other IPCPs in the same DIF to launch an attack. The N-level DIF can only see the IPC API of the N-1 DIF, so this is the only means of attacking its underlying DIF.
- The N-1 DIF transports SDUs between IPCPs in the N-level DIF, but cannot access the internals of N-level DIF.

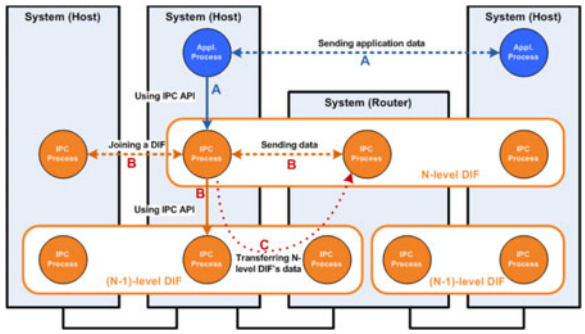


Fig. 2. Example RINA network and interaction of its components; three types of potential attacks labeled as A, B, or C.

TABLE I
RINA PAs

PAs	Type	Description
User data	Information	Any data stored on or transmitted to from an AP or IPCP, or can be inferred from such data, e.g., documents, media files.
Management data	Information	A subset of data that concerns the operation of the computing system or network, e.g., configurations, addresses, cryptographic keys, policy, measurements.
Computing resources	Service	This refers to the computer system, e.g., memory storage, processes, etc.
Communication (Com.) service	Service	The service that transfers data from one system to another.

C. Asset Identification

There are two types of assets. PAs are regarded the main resources (i.e., information and services) as the targets of an attack, which are valuable to the network and its stakeholders. A successful attack would result in damage to the PAs with an impact on the network operation. Table I provides the main PAs of RINA network.

Supporting Assets (SA) are tangible entities that enable and support the existence of PAs. Entities involved in storing, processing and/or transmitting PAs are classified as SAs. They may have vulnerabilities that can be exploited by threats targeting the PAs. Within RINA, the SAs are mostly the entities in an IPC process, as shown in Fig. 1, as well as in the management functions of DAF/DIF.

All PAs shall be linked with at least one SA, and all SAs shall be linked with at least one PA. One SA may enable several PAs, and one PA may be enabled by several SAs. Table II lists the SAs that may be targeted by a threat scenario and their related PAs. It comprises three sets of SAs: IPCP-related assets including

TABLE II
RINA SAs AND RELATED PAs

SAs	Related PAs
EFCP, IPC API, and CDAP	User management data, Com. service
RMT and flow allocation	Com. service
CACEP	Management data, Com. service
Resource allocation	Computing resources
DMS manager, enrollment, MA, authentication, and RIB daemon	Management data
KM, ACM, SDU delimiting and SDU protection	User & Management data
RIB, Local KA, AcA	User & Management data, Computing resources

protocols, DMS–related, and security-related assets that are not shown in Fig. 1 of the reference model.

V. THREAT SCENARIOS

In this paper, we only focus on intentional threats to a RINA network and its assets. Therefore, we do not analyze the complete spectrum of threats (e.g., faults, accidental, natural, criminal, terrorist damages, or unintentional misconfiguration of policies). Only the most relevant threats, according to the scope described in Section IV-B, have been selected and applied to the SAs. These threats are intended for C, I, and A violation, disruption of services, unauthorized access to data and objects, and unauthorized disclosure of information.

Accordingly, three types of potential individual attacks (labeled A, B, or C as in Fig. 2) targeting assets relevant to the DAF/DIF are considered. Here, the potential attacks are considered independently from each other, each having different scope. It should be noted that attack graphs can model how multiple related vulnerabilities may be combined for an attack [20]. Table III lists potential attacks of "A" type that originate from an AP in the DAF. We consider attacks where either the sending or receiving AP can be malicious.

In RINA, an AP only knows the destination AP-name and its own local application port identifier (ids); it never sees the network address used by the DIF. When an application requests a DIF to allocate a flow to the destination application, it provides the destination AP-name. The DIF internally resolves the destination AP-name to the address of IPCP where the destination application is registered. Therefore, an attacker cannot address an IPCP unless it joins the DIF it wants to attack. Table IV lists possible attacks of "B" type that originate within, or from an IPCP joining, the N-level DIF.

The DMS, KM, and ACM functions reside in a central point. We consider a set of threats to their associated components (i.e., MA, KA, AcA) that reside in network nodes.

Table V lists potential attacks of "C" type on the network that originate from the N-1-level DIF. We do not consider attacks within the N-1-level DIF, as we consider these to be the same as those described above.

All the identified threats in Tables III, IV, and V are subjected to risk assessment that is explained in the next section.

TABLE III
POTENTIAL "A" TYPE ATTACKS ON THE NETWORK

Threat ID	SA	Description
A1	IPC API	Attacker masquerades as another application process when requesting a flow. It calls the Allocate_Request API with source application id of another application, which is used to make access decision whether to grant request. Results in an unauthorized flow being created from the attacker to an application process.
A2	IPC API	Attacker repeatedly calls Allocate_Request API in attempt to consume resources and perform denial of service attack.
A3	IPC API	Attacker masquerades as another application process and calls the send API to inject PDUs.
A4	IPC API	Attacker repeatedly sends messages to another AP to overwhelm it and consume resources.
A5	IPC API	Attacker masquerades as another application process and calls the Deallocate API.
A6	CACEP	Attacker spoofs a RELEASE message to disrupt connection to another AP.
A7	CACEP	Attacker floods CONNECT messages to overwhelm destination and consume resources.
A8	CDAP	Malicious destination AP learns AP name and credentials of legitimate AP and uses these to impersonate the AP and join the DAF.

VI. SECURITY RISK ASSESSMENT

The approach conduced in this study for assessment and mitigation of risks is as follows.

- The level of risk to the assets is evaluated without taking into account the RINA built-in security controls.
- Then by using RINA built-in security controls and the proposed enhancements that are the subject of risk treatment process, we attempt to reduce the risk and bring the residual level of risk to low.

The risk evaluation involves the process of assigning values to the impacts and likelihood of a risk. We tailored the SecRAM methodology in two aspects in order to apply it specifically to RINA. First, SecRAM considers a number of critical impact areas to evaluate the impact on C, I, or A of PAs. These impact areas are categorized for different concerns: personnel, capacity, performance, economic, branding, regulatory, and environment [10]. We have only considered performance as the most relevant one to RINA evaluation. It is a more generic criterion than capacity and is used when a loss of C, I, or A leads to a reduction of performance of the system. Second, the impact of a threat scenario is evaluated at two levels: the inherited and reviewed. The Inherited is the impact gathered from the threat scenario given in Table VI. The reviewed impact is performed by operational staff, independent bodies, and technical teams, and is equal to or less than the inherited impact. Here, we use the inherited as the impact for each criterion.

For each threat, the impact on the C, I, and A of the network's information and services is assessed according to the following scale [10].

- Scale 1: No impact/not applicable.
- Scale 2: Minor—limited impact to the IPCP or AP, but it is still able to function.
- Scale 3: Severe—performance of the AP or IPCP is compromised.
- Scale 4: Critical—performance of the DIF or DAF is compromised.
- Scale 5: Catastrophic—RINA network or multiple DIFs are compromised.

TABLE IV
POTENTIAL "B" TYPE ATTACKS ON THE NETWORK

Threat ID	SA	Description
B1	SDU Protection	Attacker compromises SDU Protection so that it malfunctions.
B2	SDU Delimiting	Attacker compromises SDU Delimiting so that it malfunctions.
B3	RIB	Attacker accesses objects in the RIB of another IPC, e.g., key material.
B4	RIB	Attacker writes objects to the distributed RIB, e.g., changes the DIF's SDU protection policies so that SDUs are not encrypted.
B5	RIB	Malicious IPCP updates forwarding tables in RIB to route traffic through malicious nodes.
B6	RIB	Attacker deletes objects from the distributed RIB, e.g., deletes logs.
B7	IPC API	Attacker masquerades as another IPCP when requesting a flow. It calls the Allocate_Request API with source application id of another IPCP, which is used to decide whether to grant request. Results in flow being set-up between malicious IPCP and the DIF it wants to join.
B8	IPC API	Attacker repeatedly calls Allocate_Request API in attempt to consume resources and perform denial of service attack.
B9	IPC API	Attacker masquerades as another IPCP and calls the Deallocate API, causing the victim IPCP to lose its connection to the DIF.
B10	IPC API	Attacker masquerades as another AP and calls the Send API, injecting DTP messages.
B11	IPC API	Attacker repeatedly sends messages to another IPCP to overwhelm it and consume resources.
B12	CACEP	Attacker floods CONNECT messages to destination IPCP and consume resources.
B13	CACEP	Attacker spoofs a RELEASE message to disrupt connection between the targeted IPCP and DIF.
B14	Authentication	Attacker bypasses the authentication check and joins the DIF.
B15	Authentication	Attacker repeatedly sends false credentials to authentication module to consume resources.
B16	CDAP	Attacker performs a man-in-the-middle attack, intercepting CDAP packets and forwarding them on to the destination IPCP.
B17	Enrolment	Attacker masquerades another IPCP when joining DIF.
B18	Enrolment	Attacker repeatedly joins and leaves a DIF, consuming resources.
B19	Flow Allocation	Attacker compromises flow allocation so it malfunctions.
B20	Resource Allocation	Attacker compromises resource allocation so it malfunctions.
B21	RMT	Attacker compromises RMT so that it malfunctions in relaying and scheduling function.
B22	MA	Malicious IPCP compromises MA to gain access to other DIFs within a processing system.
B23	AcA	Malicious IPCP compromises AcA so it malfunctions
B24	AcA	Attacker accesses objects in the AcA store to change users' profiles.
B25	KA	Malicious IPCP compromises KA so it malfunctions.

TABLE V
POTENTIAL "C" TYPE ATTACKS ON THE NETWORK

Threat ID	SA	Description
C1 to C5	EFCP	Malicious IPCP—fabricates (C1); modifies (C2); eavesdrop (C3); replays (C4); or does not forward PDUs (C5)—received from DIF above.
C6	IPCP API	Malicious IPCP eavesdrops AP/IPCP name and credentials and uses to enroll in a DIF.
C7 to C9	CDAP	Malicious IPCP—fabricates (C7) or modifies (C8) CDAP messages, e.g., containing routing updates; or eavesdrops (C9) CDAP messages e.g., contains key material—sending them to above DIF.

TABLE VI
ASSESSED IMPACT AND LIKELIHOOD OF EACH THREAT

Threat ID	C	I	A	Overall Impact	Likelihood
A1,B7	3	1	1	3	3
A2,A7,B8,B12	1	1	3	3	3
A3,B10	2	2	1	2	3
A4,B11	1	1	4	4	3
A5,B9,A6,B13	2	1	3	3	3
A8,B4,B5	4	4	1	4	4
B1,B2	2	2	2	2	2
B3	3	3	1	3	2
B6	1	4	4	4	4
B14,B25,B23	4	4	1	4	2
B15	1	1	3	3	4
B16	3	3	1	3	3
B17,C6	4	4	1	4	4
B18	1	1	4	4	4
B19,B20	2	2	3	3	2
B21	2	2	4	4	2
B22	5	5	1	5	3
B24	4	4	1	4	3
C1,C2,C4	1	5	1	5	4
C3	5	1	1	5	4
C5	1	1	5	5	4
C7,C8	1	4	1	4	4
C9	4	1	1	4	4

The impact is valued and assessed according to the loss or degradation of C, I, and A for every PA linked to SAs (see Table II). The SAs inherit the C, I, A values of their PAs. However, in the SecRAM process: 1) when one SA enables more than one PA, it will inherit the highest C, I, A values of the PA in question; 2) when several SAs (interconnected or otherwise) enable one PA, the C, I, A criteria associated with the relevant threat/s that is linked to each SA is evaluated individually. The Overall Impact is then calculated as the highest (worst-case) of the three impact values of C, I, and A.

STAR-TRANS project [21], as an example, studied the enhancement to the risk assessment by considering that a risk incident on an asset can trigger incidents in the other assets in interconnected and interdependent transportation networks. Their approach considers: the type of initial incident/threat, identification of the interconnection between assets, and the magnitude of consequences. The triggering of incidents is modeled using an impact propagation matrix. The matrix contains "1" or "0" values in each cell indicating the triggering or not of an incident in an interconnected asset (column) caused by the threat affecting the initial asset (row). Then, the risk is estimated assuming that the security incident occurs in each triggered asset. Further details are given in [21]. In this study, we then assess and estimate the Likelihood that a threat scenario can be completed successfully and practically realized according to the following scale.

- Scale 1: Very unlikely—practically impossible.
- Scale 2: Unlikely—conceivable but unlikely.
- Scale 3: Likely—only somewhat possible.

TABLE VII
RISK LEVEL DEFINITION

	Impact				
Likelihood	**1**	**2**	**3**	**4**	**5**
5	Low	High	High	High	High
4	Low	Med.	High	High	High
3	Low	Low	Med.	High	High
2	Low	Low	Low	Med.	High
1	Low	Low	Low	Med.	Med.

- Scale 4: Very likely—quite possible.
- Scale 5: Certain—might well be expected.

Table VI shows the assessed impact and likelihood of each threat. The scoring shown in this table is subjective and depends on definition of scales above, best practices, intuition, and the security experts' knowledge. This method is opinion based but it is still one of the main methods to access the impact and likelihood by accounting for various real-world parameters. ISO/IEC standard ([7, Table. A.1], "Probability" column) introduces several techniques that are applicable for the evaluation of the likelihood of a given threat scenario. According to the SecRAM, the likelihood is built from a split into *exposure* or frequency of occurrence of the threat source and *potentiality*, the probability that once the threat source occurs, the threat scenario sequence is completed successfully. Once both likelihood layers have been evaluated, the likelihood is obtained from the average of both values.

In addition to the techniques in [7] and as an example, in [22] a mathematical approach is proposed to automatically determine the likelihood, the impact to the system, and compute the potential risk. In [23], semantic threat models are used at run-time for threat detection and diagnosis. The security expert creates a generic model to be used by the system designer for automated threat identification process. Use of machine reasoning for process automation is proposed for threat identification and assessing the run-time likelihood of threats. Some further references are given in [23] for threat and risk modeling and process automation.

It should be noted that the scoring of threats, shown in Table VI, is carried out with no consideration that built-in security controls are in place. In Section VII, we analyze these threats demonstrating that the built-in security controls in RINA reduce the calculated risks to the acceptable levels.

Once the likelihood and impact of each threat has been assessed, the level of risk can be calculated using Table VII according to the SecRAM [10]; e.g., a "Medium" risk level is defined for a likelihood scale of 4 and an Impact scale of 2.

Table VIII shows the risk level (high, medium, and low) for each of the identified threats.

VII. SECURITY CONTROLS

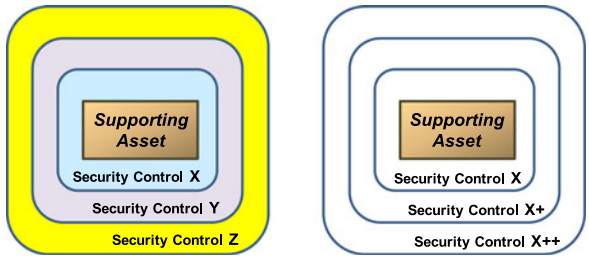
As stated in [10], security controls as treatment actions are defined to protect SAs. They are a collection of measures for managing risks and to ensure the security objectives are met. They include, but are not limited to, procedures, policies, more robust technical solutions, and management actions. The security objective levels come from the definition of the impact area as explained in Section VI. See Table IX for the performance impact area only. A security need is defined whether a risk needs to be treated or not; when the level of a risk is higher than the security objective of a SA (i.e., the lowest security objective) it is targeting, a treatment shall be applied.

TABLE VIII
CALCULATED RISK LEVEL OF EACH THREAT

Threat ID	Overall Impact	Likelihood	Risk level
A1,A2,A5,A6,A7,B3,B7,B8,B9,B12, B13,B16,	3	3	Med.
A3,B10	2	3	Low
A4,B11	4	3	High
A8,B4,B5,B6,B17,B18,C6,C7,C8,C9	4	4	High
B1,B2	2	2	Low
B14	4	2	Med.
B15,B24	3	4	High
B19,B20	3	2	Low
B21,B23,B25	2	4	Med.
B22	3	5	High
C1,C2,C3,C4,C5	5	4	High

TABLE IX
RISK APPETITE DEFINITION TABLE; SECURITY OBJECTIVE LEVEL

Security objective level	**Low**	**Medium**	**High**
Impact Impact Area	**5 or 4**	**3**	**2 or 1**
Performance	Major quality abuse making system inoperable	Severe quality abuse making system partially inoperable	Minor or No system quality abuse



(a) (b)

Fig. 3. Protection strategy approaches: (a) DiD and (b) SoC.

The risk treatment option should be selected from the actions defined in step 8 of Section II (i.e., tolerate, reduce, avoid, or transfer). Here, we choose the "Tolerate" option for the threats with "Low" risk level and the "Reduce" option in combating threats with "Medium" and "High" risk levels to meet security objective levels.

In defining the security controls, it is important to take into account the three parameters (i.e., likelihood, impact, and risk-level). For example, if the likelihood has a high value and impact has a low value, but risk level is high, the security control should be primarily defined to counter the likelihood and it could overlook the impact.

Once the type of treatment has been evaluated (e.g., for reduction of the risk), a strategy of protection for SAs should be applied in order to choose the best set of security controls. The strategy proposed by SESAR offers two approaches for combined security control: "Defense in Depth (DiD)" and "Strength of Control (SoC)"; see Fig. 3. DiD relies on a multilayered set of unrelated controls acting together to provide protection to a SA. The multilayered approach to controls means that if one

TABLE X
DEFINED SECURITY CONTROL FOR EACH THREAT WITH "MEDIUM" RISK

Threat ID	Security controls	Strategy
A1	Authenticate users of the IPC API API flow control flow allocator control access based on defined flow policy.	DiD
A2, A5, B9	Authenticate users of the IPC API.	SoC
A6	Authenticate APs when establishing a CDAP connection and include proof of authentication with CDAP messages, i.e., persist the authentication for the connection.	
A7	Authenticate APs when establishing a CDAP connection and include proof of authentication with CDAP messages, i.e., persist the authentication for the connection monitor CDAP connections MA takes actions.	DiD
B3	Control access to the RIB monitor and record access to the RIB MA takes actions.	DiD
B7	Authenticate users of the IPC API API flow control flow allocator access control.	DiD
B8	Authenticate users of the IPC API monitor and log use of the IPC API MA takes actions.	DiD
B12	Authenticate APs when establishing a CDAP connection and include proof of authentication with CDAP messages, i.e., persist the authentication for the connection monitor CDAP connections MA takes actions.	DiD
B13	Authenticate IPCPs when establishing a CDAP connection and include proof of authentication with CDAP messages, i.e., persist the authentication for the connection.	SoC
B14	Authentication must be nonbypassable and a strong authentication mechanism should be used	
B16	Authenticate the destination IPCPs when establishing a CDAP connection	
B21	Monitor RMT functions to detect abnormal behavior MA takes actions	DiD
B23	Monitor AcA to detect abnormal behavior MA takes actions	
B25	Monitor KA functions to detect abnormal behavior MA takes actions	

TABLE XI
DEFINED SECURITY CONTROL FOR EACH THREAT WITH "HIGH" RISK LEVEL

Threat ID	Security controls	Strategy
A4, B11	Authenticate users of the IPC API API flow control flow allocator access control monitor and log use of the IPC API MA/DMS takes actions.	DiD
A8	Authenticate the destination AP before sending credentials flow allocator at IPCP in N-1 DIF decides (based on policy) to reject a flow request if requester intends to overwhelmingly consume the resources.	
B4	Control access to the RIB. Only the MA is allowed to change the DIF's policies monitor and record access to the RIB MA/DMS takes actions.	
B5	Authenticate routing update messages monitor routing updates MA/DMS takes actions.	
B6	Control access to the RIB. Log files must not be deleted monitor and record access to the RIB MA/DMS takes actions.	
B15	Monitor the number of authentication failures MA/DMS takes actions in blocking the misbehaving processes that send false credentials.	
B17	Use a strong authentication mechanism.	SoC
B18	Monitor IPCPs enrolling in the DIF and leaving the DIF MA/DMS takes actions in blocking any misbehaving IPCPs.	
B22	DMS to monitor and challenges MA in random intervals DMS to detect any abnormal behavior and reconfigure MA.	
B24	Control access to the AcA data store monitor and log access to the AcA data store MA/DMS takes actions.	DiD
C1, C2	Protect the I of PDUs by using cryptographic SDU protection when sending over an untrusted N-1 DIF, signing the code for C2	SoC
C3, C6, C9	Protect the C of PDUs containing sensitive/authentication data by using cryptographic SDU protection when sending over an untrusted N-1 DIF.	
C4	Use replay protection on PDUs by using cryptographic SDU protection when sending over an untrusted N-1 DIF.	
C5	MA monitors the incoming PDUs to and outgoing from the IPCP to detect IPCPs not forwarding PDUs. MA take action (e.g., reconfiguring) against these misbehaving IPCPs.	DiD
C7, C8	Protect the I of PDUs by using cryptographic SDU Protection when sending over an untrusted N-1 DIF Verify the authenticity of CDAP messages.	DiD

control is compromised then another control should act as the next layer of defense. SoC relies on improving the performance of one "type" of control (e.g., rigorous access control, strong authorization). This means an attacker needs more expertise, better tools, and more time to break through the control.

We, therefore, identify and consider security controls for threats with a risk level of medium (see Table X) and high (see Table XI). This is to reduce the risk level to the acceptable level that corresponds to the security objective of SAs. The most feared and critical threat scenarios are with the risks evaluated as high with low security objectives. These should have high priority in treating them. Both protection strategy approaches shown in Fig. 3 are considered here to determine security controls for implementation. In most cases, layers of security controls are put in place for threats with "High" risk levels except for some where use of strong cryptographic means is advised.

There are also three implicit built-in mechanisms that strengthen the protection of assets.

1) The IPCP flow allocator in the N-1 DIF can decide based on its configured policies whether to accept a flow request or not. This can be regarded as an initial access control to the resources.
2) The IPC API in the N-DIF can block the sender and return an error if an AP or N+1 IPCP is trying to write more than is allowed to an N-flow. This API flow control affects a single AP and hence does not affect other APs, which makes it an effective guard against some of the attacks.
3) In RINA, port-ids are requested by APs for identifying application end-points and explicitly allocated during flow allocation. Port allocations are decoupled from transport connection end-points ids (cep-ids) but they are mapped via a local binding at each end-point. An instance of EFCP for performing flow control between the ceps is created by the source IPCP and identified by dynamically generated cep-ids at source and destination IPCPs. The port-ids have local significance and are used by APs to read/write data from flows while the source and destination cep-ids are used in the PDUs. This separation allows achieving greater resiliency against transport-level attacks such as port scanning and connection opening impacting communication service.

From both Tables X and XI, it can be seen explicitly that the majority of these threats are mitigated using mechanisms (i.e.,

TABLE XII
RISK LEVEL OF THREATS THAT SHOULD BE REDUCED BY MONITORING

Tc-ID	Description	Related threats
T1	An IPCP provides false information to other IPCPs.	B5
T2	An IPCP deliberately overwhelming other DIF members or the underlying DIF with messages.	A2, A4, A7, B8, B11, B12
T3	An IPCP not forwarding messages	C5
T4	An IPCP repeatedly joining and leaving a DIF.	B18
T5	An IPCP repeatedly causing errors when attempting to join a DIF.	B15
T6	Compromising the data stored in an IPCP or DIF.	B3, B4, B6, B24
T7	Compromise an IPCP or DIF function so that it malfunctions.	B21, B22, B23, B25

security controls) that are already provisioned by IPCPs and that have been considered in built-in security enablers in RINA, to satisfy the stated security requirements.

Further security controls are also provided and recommendations made for implementing some of the built-in security controls (e.g., for threat ID B14, recommends the use of non-bypassable strong authentication). To summarize, the security controls can be categorized as follows.

- Authenticate both the source and destination APs or IPCPs when establishing a CDAP connection.
- Authenticate users of the IPC API. Note that this may be implemented in the system's operating system.
- Controlling access to the resources via access control.
- Using SDU protection to protect the C and I of PDUs.

The above security control mechanisms require the services of either a KM, ACM, or both and their associated components to be integrated as part of the function of RINA at system, DIF, or management level.

The remaining threats can be reduced by performing monitoring of activities to identify processes that are not behaving as expected and when to take actions against them. These threats can be generalized into the threat categories (Tc-ID) described in the Table XII.

VIII. PROTOTYPING AND SECURITY CONTROLS

RINA's network operation is dictated by policy to achieve more flexibility and provide ease of configuration. A software development kit (SDK) has been developed [24] in the PRISTINE project to put on top of the IRATI open-source prototype [25] in order to support DIF programmability. This allows customization of the policies and reuse of the same IPCP mechanisms across DIFs adapting the network to different operating environments. The SDK is an open-source software and is publicly available at the GitHub repository [26]. Further SDK versions will also be released allowing the use of any newly implemented components, policy plug-in, and dynamic policy selection during different project's phases.

In Table XII we identified seven threats that should be monitored for in a DIF to detect runtime attacks. We, therefore, need to ensure that the relevant specifications are defined for the managed objects for monitoring these threats. The IPCP's security management component and MA at the local-level or DMS manager at the domain-level can maintain realistic courses of action through the defined policies that proactively or reactively address cyber-attacks. In response to the monitoring indications (e.g., detection of divergence from conditions of normal operations using different security metrics) that an attack is underway the DMS manager, or MA on its behalf, can take actions. An overview of metrics is given in [27] for different aspects namely reliability, security, and performance.

Here, we take one of the threats (T4 in Table XII) as an example to present the managed object model definition and instantiate monitoring actions. In this threat one or more IPC processes repeatedly join and leave a DIF in order to consume resources. The first step is for the manager to configure the MA to monitor the number of successful IPCP attempts to join a DIF. It does this by requesting the MA to create a forwarding discriminator that will notify the manager when a number of successful IPCP attempts are made to enroll in the specified DIF. The discriminator then filters the notifications from the IPC process. When it has received the specified number of notifications for successful IPCP attempts to enroll in the DIF, the MA sends a report to the manager. It is left to the manager using its defined policies to decide how to deal with any excessive attempts to join the DIF. Based on the attack characteristics, the MA, or DMS manager can take actions, i.e., changes to policies, reducing the operation of IPCP processes, or modifying the security postures of component/s under attack to launch an adaptive response for withstanding or recovering from cyber threats.

IX. FORMAL ANALYSIS OF RINA SECURITY CONTROLS

This section describes formal analysis experiments that deal with threats related to communication between IPC processes. The experiments consist of a formal analysis of identified security risks followed by the demonstration of found flaws using a simulation model. Among many possible threats, we only consider attacks related to IPCP communication. Dolev-Yao model [28] of the attacker can be applied to analyze these attacks. All these attacks assume that the attacker has access to a channel that carries traffic between communicating parties. In the case of RINA, this channel can be a physical communication medium in a ShimDIF, or a malicious IPCP that offers its services to DIF above. The complexity of creating a testbed environment for verification of defined security controls in protecting RINA assets led us to consider alternative means of verification. Formal methods are mathematically based techniques for the specification, development, and verification of software aspects of digital systems. We have considered formal verification techniques and applied the ProVerif tool[1] for formal analysis of RINA security and RINASim [29] for a demonstration of identified security threats. Applying a formal tool for verification of security mechanism enables us to determine the attack traces and verify the properties of security measures applied to mitigate the security threats associated with these attacks. The results presented mainly comprise of formally verified RINA network architecture with respect to security. This work can be viewed as a complementary analysis to that done by [30].

A. RINA Network Model

In this section, we will build a model similar to Fig. 2 that integrates an abstraction of the RINA process behavior related to the attacks being analyzed. The aim is to show the attacks described in the previous sections and how these attacks can be avoided by applying the cryptographic SDU Protection policy.

[1] [Online]. Available: http://prosecco.gforge.inria.fr/personal/bblanche/proverif/

In our formal model, the host is assigned the following names for their easy referencing: ALICE represents the left-hand side host running RINA client application, and BOB represents a right-hand side host running the RINA server application. Also, the router node is denoted by the name DAVE. We built the formal model using ProVerif.

B. Analysis of Attacks and Security Controls

Potential attacks of "A" type on the network that originate from an AP in the DAF where either the sending or receiving AP can be malicious. Possible attacks of "B" type that originate within the N-level DIF or from an IPCP joining the N-level DIF. Potential attacks of "C" type on the network that originate from the N-1-level DIF. The presented formal model considers "C" type attacks. We begin with formal definitions of security properties related to these attacks. Malicious IPCP in (N-1) DIF can be located at the ALICE or DAVE host. For analysis, however, it is possible to assume that an attacker has access to the channel in (N-1) DIF (referenced as red DIF in our formal model). As the attacker can listen to the channel, insert new data into the channel, or modify any data on this channel, it is possible to analyse a large subset of identified attacks. The following definitions specify the assertions that express the attacks formally using the ProVerif language.

Definition 1: Attack C1—Malicious IPCP fabricates PDUs from DIF above.

An attacker creates PDU and sends it to the victim IPC process. The attack is successful if the victim IPCP accepts the PDU fabricated by the attacker. The security property that claims the impossibility of this attack is expressed as the following correspondence assertion.

$$\textbf{query}\ p{:}Pdu;\ \textbf{event}(PduAcceptEvent(\mathsf{BOB\_BLUE}, p)) \Rightarrow \textbf{event}(PduSendEvent(\mathsf{ALICE\_BLUE}, p)).$$

This correspondence claims that the message, to be accepted by the receiver, has to be sent by the actual sender.

Definition 2: Attack C2—Malicious IPCP modifies PDUs from DIF above.

This attack is successful if the attacker can send a modified PDU to a receiver and the receiver accepts this PDU. We specify this assertion with the help of CDAP_MESSAGE, which stands for any valid CDAP message in our model.

$$\begin{aligned}&\textbf{query}\ sp{:}SduProtection;\\ &\ \textbf{let}\ pdu = MakeDataPdu(\mathsf{BLUE\_DIF}, \mathsf{ALICE\_BLUE}, \mathsf{BOB\_BLUE}, MakeDafSdu(sp, \mathsf{CDAP\_MESSAGE}))\\ &\ \textbf{in}\ \textbf{event}\ (PduAcceptEvent(\mathsf{BOB\_BLUE}, \mathsf{ModifyPdu}(pdu))).\end{aligned}$$

Definition 3: Attack C3—Malicious IPCP eavesdrops PDU from DIF above.

This attack is done by observing the communication and capturing the content of PDU from N-DIF. The corresponding security property claims that an attacker cannot get the content of the PDU. This is again expressed with the help of CDAP_MESSAGE object as follows:

$$\begin{aligned}&\textbf{query}\ sp{:}SduProtection;\\ &\ \textbf{let}\ pdu = MakeDataPdu(\mathsf{BLUE\_DIF}, \mathsf{ALICE\_BLUE}, \mathsf{BOB\_BLUE}, MakeDafSdu(sp, \mathsf{CDAP\_MESSAGE}))\\ &\ \textbf{in}\ attacker(pdu).\end{aligned}$$

Definition 4: Attack C4—Malicious IPCP replays PDUs from DIF above.

The replaying attack is performed by sending the same PDU to the target IPCP. The security property is expressed using injective correspondence. This correspondence claims that each accepted PDU is sent by the legitimate sender.

$$\begin{aligned}&query\ sp{:}SduProtection;\\ &\ let\ pdu = MakeDataPdu(BLUE\_DIF, ALICE\_BLUE, BOB\_BLUE, MakeDafSdu(sp, CDAP\_MESSAGE))\\ &\ in\ inj\text{-}event\ (PduAcceptEvent(BOB\_BLUE, pdu))\\ &\ ==>\ inj\text{-}event\ (PduSendEvent(ALICE\_BLUE, pdu)).\end{aligned}$$

Definition 5: Attack C7—Malicious IPCP fabricates CDAP message.

This attack is accomplished by creating a CDAP message such that the DAF IPCP accepts this message. It does not necessarily mean the application itself agreed to the information of the message. However, the content of fabricated CDAP message is consumed by the application. The security property that corresponds to this attack is given by the following correspondence:

$$\textbf{query}\ m{:}Cdap;\ \textbf{event}(CdapReceiveEvent(\mathsf{BOB\_SERVER}, m)) \Longrightarrow \textbf{event}(CdapSendEvent(\mathsf{ALICE\_CLIENT}, m)).$$

This correspondence expresses that accepted CDAP messages have to be sent by the legitimate source application only. Violating this correspondence reveals the attack trace.

Definition 6: Attack C8—Malicious IPCP modifies CDAP messages.

This attack is successful if the attacker can send a modified message to a receiver and the receiver accepts this message. To state this formally, CDAP_MESSAGE stands for all legitimate CDAP messages and the ModifyCdap function is used to represent an operation that takes an original message and produces a (modified) cdap message. Event CdapReceiveEvent signalizes that the receiver gets the modified message.

$$\textbf{query}\ \textbf{event}(CdapReceiveEvent(\mathsf{BOB\_SERVER}, \mathsf{ModifyCdap}(\mathsf{CDAP\_MESSAGE}))).$$

The existence of a state reachable in the network model that satisfies this assertion makes this attack possible.

Definition 7: Attack C9—Malicious IPCP eavesdrop CDAP message.

This attack is successful if the attacker can read a CDAP message with possible sensitive data. To state this formally in ProVerif, this attack is represented as the following query:

$$\textbf{query}\ attacker(\mathsf{CDAP\_MESSAGE}).$$

Here, CDAP_MESAGE stands for any message sent by the RINA application process. Thus, finding any state in which the attacker may read this message represents the possibility to deploy the attack C9.

Next, we present theorems that express security properties of the RINA under the different security policies applied. First, we prove the correctness of the model by showing that legitimate communication can be delivered.

Theorem 1: Legitimate CDAP message can be delivered to the receiver.

Proof: ProVerif can automatically analyze this property by proving the following correspondence:

```
   event(CdapReceiveEvent(BOB_SERVER,
CDAP_MESSAGE)) ==>
event(CdapSendEvent(ALICE_CLIENT,
CDAP_MESSAGE))
```

This correspondence expresses that CDAP_MESSAGE can only be accepted by the receiver if the legitimate sender previously sent it. The next theorem reveals the vulnerability of RINA if strong SDU protection policy is not applied. Here the weak SDU protection policy is modeled as SDUP_NONE.

Theorem 2: RINA is vulnerable to C-type attacks when default SDU protection policy is applied.

Proof: ProVerif falsifies the following security properties:

```
   RESULT not
event(CdapReceiveEvent(BOB_SERVER,ModifyCdap(
CDAP_MESSAGE[]))) is false.

   RESULT
event(CdapReceiveEvent(BOB_SERVER,m)) ==>
event(CdapSendEvent(ALICE_CLIENT,m)) is
false.

   RESULT not attacker_Cdap(CDAP_MESSAGE[])
is false.

   RESULT not
attacker_Pdu(MakeDataPdu(BLUE_DIF,ALICE_BLUE,
BOB_BLUE,MakeDafSdu(sp_4754,CDAP_MESSAGE[])))
is false.

   RESULT event(PduAcceptEvent(BOB_BLUE,p))
==> event(PduSendEvent(ALICE_BLUE,p)) is
false.
```

ProVerif is also able to find a trace for each of these security properties. It means that attacks are possible, and thus, RINA is vulnerable to these attacks. Finally, we prove the theorem claiming that the RINA is protected to "C" type threats by application of the Crypto SDU protection policy.

Theorem 3: Crypto SDU protection policy protects RINA against C1–C9 attacks.

Proof: ProVerif automatically proves the following security properties:

```
   RESULT not
event(CdapReceiveEvent(BOB_SERVER,ModifyCdap(
CDAP_MESSAGE[]))) is true.

   RESULT
event(CdapReceiveEvent(BOB_SERVER,m)) ==>
event(CdapSendEvent(ALICE_CLIENT,m)) is true.

   RESULT not attacker_Cdap(CDAP_MESSAGE[])
is true.

   RESULT not
attacker_Pdu(MakeDataPdu(BLUE_DIF,ALICE_BLUE,
BOB_BLUE,MakeDafSdu(sp_4164,CDAP_MESSAGE[])))
is true.

   RESULT not
event(PduAcceptEvent(BOB_BLUE,ModifyPdu(MakeD
ataPdu(BLUE_DIF,ALICE_BLUE,BOB_BLUE,MakeDafSd
u(sp_5465,CDAP_MESSAGE[])))))) is true.

   RESULT event(PduAcceptEvent(BOB_BLUE,p))
==> event(PduSendEvent(ALICE_BLUE,p)) is
true.

   RESULT
event(CdapReceiveEvent(BOB_SERVER,CDAP_MESSAG
E[])) ==>
event(CdapSendEvent(ALICE_CLIENT,CDAP_MESSAGE
[])) is true.
```

By proving these properties, the ProVerif claims that the traces of attacks cannot be found in the model. Assuming perfect cryptography, we can conclude that an attacker cannot successfully implement the C1–C9 attacks against the host IPC processes in the protected DIFs. The presented analysis deals with a high-level model of RINA. This analysis does not include any implementation details of SDU protection. This analysis is aimed at formal verification of the high-level design of SDU Protection security control. Extending the analysis to cope with more details is left for future work.

X. Conclusion

In this paper, we particularly focused on the use of a SecRAM for network systems. We performed a study to identify and prioritize run-time threats to the RINA network. Using this methodology step-by-step, we identified possible threats to RINA components, accessed the risk levels related to these threats, and identified the security controls to bring the high and medium risk levels down. We evaluated the existing RINA security controls and proposed some additional functions. Through evaluation of the risk related to the identified potential attacks, we functionally established that the architectural structure of RINA provides an inherent secure environment and this can be further enhanced by considering the stated recommendations in the implementation of embedded security mechanisms. We also established that some of the threat scenarios require monitoring of the IPCPs specific activities during network operation for reducing the threats' risk levels. In order to realize the security state of RINA's network system, monitoring should be carried out for observing and gathering data from different indicators, processing events, identifying adversary activities, and, thus, discovering the needs for any adaptation of IPCP/DIF policies. Work conducted in the European Commission FP7 PRISTINE and its follow up H2020 ARCFIRE projects are to further specify the relevant policies. In addition, for proof of concept, the IRATI implementation and the developed SDK are used to further evaluate the operation of RINA along with security controls using the analysis reported here in three use-cases namely distributed cloud, data centre networking, and network service provider [19].

We provided a formal analysis of selected attacks on RINA network hosts and communication. We applied the ProVerif tool to create a formal model of a RINA network and selected attacks. Employing capabilities of ProVerif, we were able to formally prove that by applying SDU protection policy, these attacks can be mitigated. Among the analyzed attacks, the replay attack has led to the most interesting security properties. While the output from ProVerif is easy to understand, for better explanation of attacks the simulation model using RINASim was developed. The simulation runs presented traces found by the ProVerif that correspond to the analyzed attacks. The scope of the presented analysis is limited by the Dolev–Yao model of the attacker, implemented by the ProVerif. With this, only high-level security properties of RINA architecture can be analyzed. For detailed analysis of SDU protection functions it is more suitable to develop a computation model and use the CryptoVerif tool,[2] for instance. The presented approach also demonstrated that formal analysis of security properties of a real-network architecture is possible when supported by available formal tools.

[2][Online]. Available: http://prosecco.gforge.inria.fr/personal/bblanche/cryptoverif/

ACKNOWLEDGMENT

The authors would like to thank their project colleagues for the fruitful discussions and comments made in the development of work presented in this paper.

REFERENCES

[1] D. J. Bodeau and R. Graubart, "Cyber resiliency assessment: Enabling architectural improvement," MITRE Technical Report, May 2013. [Online]. available: http://www.mitre.org/publications/

[2] J. Day, *Patterns in Network Architecture: A Return to Fundamentals*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2008.

[3] *Guidance for Establishing the Context*, ISO/IEC 27005, 2008. [Online]. Available: www.pqm-online.com/assets/files/lib/std/iso_iec_27005-2008.pdf

[4] National Institute of Standards and Technology (NIST), "Guide for conductig risk assessment," Special Publication 800-30 Rev. 1, Sep. 2012. [Online]. Available: http://dx.doi.org/10.6028/NIST.SP.800-30r1

[5] The MITRE Institute, "Risk management," System Engineering Guide, Sep. 2013. [Online]. Available: http://www.mitre.org/publications/

[6] European Network and Information Security Agency (ENISA), "Risk management: Implementation principles and inventories for risk management/risk assessment methods and tools," Technical Report conducted by the Technical Department of ENISA Section Risk Management, Jun. 2006. [Online]. Available: https://www.enisa.europa.eu/publications/

[7] *Risk Management-Risk Assessment Techniques*, International Organization for Standardization & International Electrotechnical Commission ISO/IEC 31010, Geneva, Switzerland, 2009, [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iec:31010:ed-1:v1:en

[8] F. Hermann and D. Khadraoui, "Security risk management methodologies," in *Advances in Enterprise Information Technology Security*. Calgary, AB, Canada: IGI Global, 2008, pp. 261–273, Ch. 14.

[9] A. Ekelhart, S. Fenz, and T. Neubauer, "AURUM: A framework for information security risk management," in *Proc. 42nd Hawaii Int. Conf. Syst. Sci.*, Jan. 2009, pp. 1–10.

[10] SESAR Joint Undertaking, "SESAR ATM SecRAM implementation guidance material," Project 16.02.03 D03, 2013. [Online]. Available: http://www.sesarju.eu/

[11] H. Asgari, S. Haines, and A. Waller, "Security risk assessment and risk treatment for integrated modular communication," in *Proc. Int. Conf. Availability, Reliab. Security*, Salzburg, Austria, Sep. 2016, pp. 503–509.

[12] A. Marotta, G. Carrozza, L. Battaglia, P. Montefusco, and V. Manetti, "Applying the SecRAM methodology in a cloud-based ATM environment," in *Proc. 8th Int. Conf. Availability, Reliab. Security*, Regensburg, Germany, Sep. 2013, pp. 807–813.

[13] S. M. Bamakan and M. Dehghanimohammadabadi, "A weighted Monte Carlo simulation approach to risk assessment of information security management system," *Int. J. Enterprise Inf. Syst.*, vol. 11, no. 4, pp. 63–78, Oct.–Dec. 2015.

[14] P. Saripalli and B. Walters, "QUIRC: A quantitative impact and risk assessment framework for cloud security," in *Proc. IEEE 3rd Int. Conf. Cloud Comput.*, Miami, FL, USA, Jul. 2010, pp. 280–288.

[15] G. Kalogridis, M. Sooriyabandara, Z. Fan, and M. A. Mustafa, "Toward unified security and privacy protection for smart meter networks," *IEEE Syst. J.*, vol. 8, no. 2, pp. 641–654, Jun. 2014.

[16] E. Grasa, O. Rysavy, O. Lichtner, H. Asgari, J. Day, and L. Chitkushev, "From protecting protocols to protecting layers: Designing, implementing and experimenting with security policies in RINA," in *Proc. IEEE Int. Conf. Commun.*, Kuala Lumpur, Malaysia, May 2016, pp. 736–742.

[17] G. Boddapati J. Day, I. Matta, and L. Chitkushev, "Assessing the security of a clean-slate internet architecture," in *Proc. 20th IEEE Int. Conf. Netw. Protocols*, Austin, TX, USA, 2012.

[18] J. Small, J. Day, and L. Chitkushev, "Threat analysis of recursive internetwork architecture distributed inter-process communication facilities," Boston University Technical Note, Boston, MA, USA. 2011. [Online]. Available: http://rina.tssg.org/docs/js-001.6-RINA_Threat_Analysis.pdf

[19] PRISTINE (programmability in RINA for European supremacy of virtualised networks) project deliverables, Jan. 2014–Oct. 2016. [Online]. Available: http://ict-pristine.eu/

[20] S. Noel, S. Jajodia, L. Wang, and A. Singhal, "Measuring security risk of networks using attack graphs," *Int. J. Next-Gener. Comput.*, vol. 1, no. 1, pp. 135–147, Jul. 2010.

[21] STAR-TRANS EC Project, "Strategic risk assessment and contingency planning in interconnected transport networks," Final Publishable Summary Rep., Sep. 2012. [Online]. Available: http://cordis.europa.eu/result/rcn/140443_en.html

[22] T. Olsson, "Assessing security risk to a network using a statistical model of attacker community competence," *Inf. Commun. Security*, vol. 5927, pp. 308–324, 2009.

[23] M. Surridge *et al.*, "Run-time risk management in adaptive ICT systems," in *Proc. 8th Int. Conf. Availability, Reliab. Security*, Regensburg, Germany, Sep. 2013, pp. 102–110.

[24] V. Maffione, F. Salvestrini, E. Grasa, L. Bergesio, and M. Tarzan, "A software development kit to exploit RINA programmability," in *Proc. IEEE Int. Conf. Commun.*, Kuala Lumpur, Malaysia, pp. 316–322, May 2016.

[25] S. Vrijders *et al.*, "Prototyping the recursive internet architecture: The IRATI project approach," *IEEE Netw. Mag.*, vol. 28, no. 2, pp. 20–25, Mar./Apr. 2014.

[26] The (open) IRATI's stack. [Online]. Available: https://github.com/IRATI/stack. Accessed on: Jul. 2017.

[27] I. Eugeld, F. C. Freiling and R. Reussner, *Dependability metrics: Advances Lectures LNCS 4909*. Berlin, Germany: Springer-Verlag, 2008.

[28] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[29] RINA Simulator, an OMNeT++ feature project developed in PRISTINE. [Online]. Available: https://rinasim.omnetpp.org/. Accessed on: Jul. 2017.

[30] V. Vesely, M. Marek, T. Hykel, and O. Rysavy, "Skip this paper-RINASim: Your recursive internetwork architecture simulator," in *Proc. 2nd OMNeT++ Community Summit*, 2015, pp. 1–5. [Online]. Available: http://arxiv.org/abs/1509.03550

Hamid Asgari (SM'03) received the Ph.D. degree in electrical and electronics engineering from the University of Wales, Swansea, U.K., in 1997.

He has been with Thales UK Research, Technology, and Innovation, Reading, U.K., since 1996, where he has been a Chief Technical Consultant. He has significant experience in international research collaborations and in leading large research and development teams. He is the Network Research Lead working in advanced and future networking and security concepts, wired/wireless networks architectures and technologies, network and service management, and network risk and performance evaluations. Since 2014, he has been a Visiting Professor with King's College London. He has authored or co-authored 60 book chapters and papers in scientific journals and peer-reviewed conferences.

Prof. Asgari is an IET Fellow and a Senior Member of the ACM.

Sarah Haines received the M.Eng degree (with honors) from the University of Cambridge, Cambridge, U.K. in 2007.

She joined Thales UK Research, Technology and Innovation, Reading, U.K., in 2007, where she is currently a Principal Engineer. Her research interests include security architectures, multilevel security, identity and access management, content-based security, and advanced cryptography.

Mrs. Haines is a Certified Information Systems Security Professional and a member of the Institution of Engineering and Technology, U.K.

Ondrej Rysavy received the Ph.D. degree in computer science from the Brno University of Technology, Brno, Czech Republic, in 2005.

He is an Associate Professor with the Department of Information Systems, Brno University of Technology, Brno. His research interests include computer networking and, in particular, network monitoring, network security and forensics, and network architectures. His work is focused on improving network security through data analysis by application of data mining, statistics, and distributed computing.