An Assurance-Based Risk Management Framework for Distributed Systems

Marco Anisetti, Claudio A. Ardagna, Nicola Bena
Department of Computer Science
Università degli Studi di Milano
Milan, Italy
{firstname.lastname}@unimi.it

Andrea Foppiani
Technology Governance Risk and Control
Cargill Inc.
Milan, Italy
andrea_foppiani@cargill.com

***Abstract*—The advent of cloud computing and Internet of Things (IoT) has deeply changed the design and operation of IT systems, affecting mature concepts like trust, security, and privacy. The benefits in terms of new services and applications come at a price of new fundamental risks, and the need of adapting risk management frameworks to properly understand and address them. While research on risk management is an established practice that dates back to the 90s, many of the existing frameworks do not even come close to address the intrinsic complexity and heterogeneity of modern systems. They rather target static environments and monolithic systems thus undermining their usefulness in real-world use cases. In this paper, we present an assurance-based risk management framework that addresses the requirements of risk management in modern distributed systems. The proposed framework implements a risk management process integrated with assurance techniques. Assurance techniques monitor the correct behavior of the target system, that is, the correct working of the mechanisms implemented by the organization to mitigate the risk. Flow networks compute risk mitigation and retrieve the residual risk for the organization. The performance and quality of the framework are evaluated in a simulated industry 4.0 scenario.**

***Index Terms*—Risk Management; Assurance; Network Flows; Security; Testing**

I. Introduction

From the original launch of cloud computing and IoT, more than 10 years ago, technology has rapidly progressed and several aspects of cloud/IoT came to maturity entering commercial offerings. Current systems increasingly move from a centralized approach where computations are done at the core of the network, to a hybrid scenario where analytics and knowledge extraction are partially done at the edge near the physical environment and sensors where data are collected. A wealth of new services in different domains, such as smart vehicles, smart buildings, e-health, are distributed on the basis of data collected and computations done by devices.

The huge step forward in terms of new applications and services does not found a counterpart in non-functional aspects, being security and safety still bound to solutions that are proper of traditional distributed systems [1]. This scenario introduces new fundamental risks and the need of revisiting existing risk management frameworks to accomplish the peculiarities of modern environments. Some frameworks have been developed through the years [2], though most of them fall short in tightly integrating adequate methodologies supporting the claims at the basis of risk evaluation in modern environments. They are in fact designed for static environments, where risk mitigation is done once and never refined, mostly targeting monolithic systems with fixed architectures. They implement a risk assessment process that is more an art than a science, impairing its effectiveness in complex and dynamic environments. In other words, risk must be properly validated to avoid a false sense of security [3] and incidents with disastrous consequences [4].

Our paper aims to fill in the above gaps by defining an assurance-based risk management framework. The framework implements a qualitative and adaptive process grounded on standard risk management practices, supporting an "helicopter view" of the organization risk posture against its risk requirements. It is centered around the concept of non-functional properties as the primary source of requirements. It also provides an assurance-based approach where the overall risk is weighted on the specific behavior of each mechanism implemented by the organization to mitigate it. In a nutshell, our approach supports a process where large and complex organizations can evaluate their risk on the basis of the assurance results regarding the strength of the implemented mechanisms. It also supports the organization compliance against internal and external stakeholder expectations.

The contribution of this paper is threefold. We first extend standard risk management frameworks with an approach centered around non-functional properties, where properties model the expected behavior of the system under evaluation for risk mitigation. We then model the problem of risk mitigation as a flow network. We finally introduce an end-to-end approach integrating risk management and assurance techniques. The latter are used to monitor the status of the implemented mechanisms and their properties, thus supporting run-time risk evaluation.

The remainder of this paper is organized as follows. Section II presents the requirements for a risk management framework for modern distributed environments and our reference scenario. Section III introduces our risk management framework, while Section IV presents the flow network-based methodology for risk assessment and mitigation. Section V describes the integration of assurance techniques within the

framework in Sections III and IV. Section VI presents an experimental evaluation of our approach, discussing its performance and quality in our reference scenario. Section VII describes related work. Finally, Section VIII draws our conclusions.

II. Requirements and Reference Scenario

A. Requirements

The advent of cloud, edge, and IoT has revolutionized the architecture of distributed systems and the working of their processes [5], pushing for new security and safety approaches. Among them, the role of risk management stands out as a means to increase system robustness and user awareness, on one side, and to drive a cost-effective strategy for minimizing the impact of the observed risks, on the other side. Based on the work in [1], [3], [6], we identify the main requirements a risk management framework for modern distributed environments has to address (MUST/SHOULD) as follows.

- **Standardized approach**: it MUST follow well-established standards, guidelines, and best practices (e.g., ISO 31000 [7], NIST SP800-30 [8]).
- **Consistent and unified**: it MUST adopt consistent processes within a comprehensive and unified framework, ensuring that risk is managed effectively, efficiently, and coherently across an organization.
- **Abstraction**: it SHOULD support the abstraction of entities involved in the risk management process [1]. In turn, it SHOULD unleash a *general* approach not bound to any particular domains, permitting a wide applicability.
- **Scalability**: it MUST support a scalable qualitative risk management regardless of the size and complexity of the organization.
- **Automation**: it SHOULD support parametrization and automation of different phases of its execution [1].
- **Ranking**: it MUST provide simple yet intuitive indicators measuring the results of the risk management framework with respect to the risk criteria of the organization.
- **Assurance integration**: it MUST integrate the risk management process with assurance techniques. Assurance techniques evaluate the effectiveness of the countermeasures operated by the organization to minimize the risk and enable a *realistic* view thereof [3].
- **Continuous Process**: it SHOULD support a continuous risk management process, enabling prompt reactions to any change in the organization and implemented countermeasures.
- **Propagation**: it SHOULD manage *risk propagation* between resources, under the assumption that, in case an adverse event happens, its impact propagates to different resources [6].

To the best of our knowledge, no risk management frameworks address the above requirements (see Section VII for a detailed analysis). In other words, there are no standardized and consistent approaches retrieving a ranked and assurance-based realistic view of risk in organizations of any size.

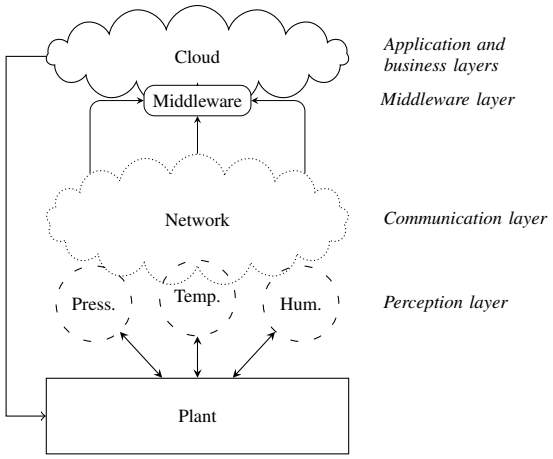

Figure 1. Reference Scenario Architecture.

B. Reference Scenario

Our reference scenario is an industry 4.0 manufacturing company, implementing the 5-tier IoT architecture in Figure 1 [5], [9], [10]. The manufacturing company deploys a number of sensors and actuators, such as temperature, humidity, pressure, infrared, and proximity, at the *perception layer*. These devices collect relevant data in their proximity and forward them to other layers for processing. The communication infrastructure is built at the *network layer* (also known as communication layer). It is based on traditional networking technologies (e.g., 3G, 4G, 5G, WiFI, infrared) and supports the transfer of data collected at the *perception layer* to the *middleware layer*. The latter is where data are persisted, transformed, and analyzed. The *application layer* then runs the manufacturing process on the basis of the data received by the *middleware layer*. The *business layer* finally manages the manufacturing process analyzing collected results and planning improvements on the overall process. *Business*, *application*, and *middleware layers* are deployed in the cloud.

Security and safety are fundamental requirements in this scenario, including the traditional *Confidentiality, Integrity, Availability* (CIA) triad. The organization must ensure confidentiality of those sensors data (e.g., pressure) that are a company trade secret. The organization must then ensure integrity of sensors data (e.g., temperature and humidity), to guarantee a proper monitoring of the manufacturing environment and ensure high quality of the products. Finally, the organization must ensure system and data availability, to enable proactive maintenance of equipments, avoiding unexpected downtime and physical harm to personnel of the organization.

Risk assessment and treatment in our reference scenario must be driven by the requirements in this section. Given the complexity of the system, they require a scalable framework, prone to automation and consistent in all steps, without forcing to particular choices. The organization must also trust the results, meaning that rankings produced by the framework ground on facts and informed decisions can be taken over them.

Table I
TERMINOLOGY

Term	Symbol	Description
Asset	A	An asset, tangible or intangible, that is valuable for the organization [11]
Non-functional property	p	A non-functional requirement to be proved on an asset
Impact	$\mathcal{I}_{(A,p)}$	A function expressing the impact of the lack of p on A
Threat	t	An event with undesired consequences [11]
Likelihood	$\mathcal{L}^t_{(A,p)}$	A function expressing the likelihood of t to become an actual attack against (A, p)
Risk	$\langle A, p, t\rangle$	A risk affecting the organization
Risk value	$\mathcal{R}^t_{(A,p)}$	The value of risk $\langle A, p, t\rangle$
Total risk per asset and property	$\mathcal{R}_{(A,p)}$	The sum of the risk values affecting A under p
Total risk per property	$\mathcal{R}_p$	The sum of the risk values under p
Risk treatment		The process of deciding what to do with the identified risks
Risk mitigation		A risk treatment approach, where risks are mitigated by mechanisms
Mechanism	m	A countermeasure mitigating a risk
Mechanisms mitigating risk	$\theta(A, p)$	The set of mechanisms mitigating risk affecting (A, p)
Expected mitigation degree	$\delta_p(m)$	A function expressing the degree to which m is expected to mitigate a risk under p
Flow network	$\mathcal{G}_p$	The flow network for property p
Mitigated risk		The amount of risk an organization can mitigate under p
Maximum flow	f	The maximum flow passing on $\mathcal{G}_p$, corresponding to the mitigated risk under p
Residual risk		The amount of risk an organization cannot mitigate for p, corresponding to the difference between the sum of the assets' total risk and mitigated risk for p
Property hierarchy	$(\mathcal{P}, \succeq_p)$	A hierarchy representing the possible configurations of non-functional properties
Assurance Model	$\langle p, m, \{tc_i\}\rangle$	A model driving the evaluation activities aimed to prove support for p on m according to test cases $\{tc_i\}$
Adjusted mitigation degree	$\delta_p(m)'$	The adjusted risk mitigation degree of m under p resulting from assurance evaluation
Adjusted flow network	$\mathcal{G}'_p$	Adjusted flow network for property p

III. RISK MANAGEMENT FRAMEWORK

Existing risk management frameworks (e.g., [7]) mostly implement static processes, whose activities are either asset-based or threat-based [1]. The approach in this paper extends existing solutions, implementing a risk management framework that addresses the requirements in Section II-A. It departs from the assumption that risk evaluation is built on the types of countermeasures adopted to mitigate the threats on the assets, rather considering the strength of the specific mechanisms integrated in the system. Our risk management framework covers both asset and threat assessments, while being centered around the concept of non-functional properties. Non-functional properties model the behavior of the system under evaluation and are bound to specific mechanisms used to take the risk under control. We note that our framework is generic and not bound to any particular implementation, and permits to fine-tune every aspect of the process according to the considered environment. It follows standards and best practices, implementing a four-phases process: *i) asset assessment*, identifying the assets and evaluating the corresponding *impact*; *ii) threat assessment*, identifying the threats and evaluating the corresponding *likelihood*; *iii) risk calculation*, qualitatively evaluating the risk based on *i)* and *ii)*; *iv) risk treatment*, determining residual risk. Table I presents the terminology used throughout this paper.

A. Asset Assessment

Phase asset assessment first identifies all relevant assets for the organization. An asset can be anything the organization thinks is valuable, tangible or intangible, as for instance, information assets [11]. It then links each asset to the non-functional properties it should preserve thereon. Non-functional properties indirectly model an expectation on the asset strength, for instance, confidentiality, integrity, and availability, and are proven on specific mechanisms implemented by the system under assessment. For instance, an asset can be a sensor data, a property for the asset can be confidentiality, an encrypted storage can be a mechanism used to link the property to the asset. This phase defines a set $\{(A_i, p_j)\}$, where A_i is an asset and p_j is a property of interest on A_i. Each pair (A_i, p_j) is assigned with a score, named *impact*, as follows.

Definition 1 (Impact): Let A_i be an asset and p_j be a property. The impact of the lack of p_j on A_i is defined by function $\mathcal{I}_{(A_i,p_j)}$: $\{A_1, \ldots, A_n\}\times\{p_1, \ldots, p_m\}\rightarrow\mathbb{N}$.

The higher the impact, the higher the damage an organization would face in case p_j is violated on A_i. There are several ways to conduct an asset assessment, for instance, some security standards require to keep an inventory or a catalog [12], [13]. Typically, it assesses several factors including business continuity disruption, financial loss, reputation damage, violations of laws, contracts or industry regulations, health and safety [14]. Our framework is generic and can support any approaches to asset assessment.

Example 1: Let us consider our reference scenario in Section II-B. Data collected from sensors, for instance, *humidity* and *temperature* (A_{ht}), are important assets of the manufacturing company. Properties *integrity* (p_I) and *availability* (p_A) should then hold, since their lack could result in a very high damage. Assuming the impact to take values in $[1, 5]$, $\mathcal{I}_{(A_{ht},p_I)}$=4 and $\mathcal{I}_{(A_{ht},p_A)}$=4. We note that *confidentiality* (p_C) is less relevant since the correct values of A_{ht} are well known in the field, as opposite to *pressure* (A_p) that is a trade secret.

B. Threat Assessment and Risk Calculation

Phase threat assessment and risk calculation identifies all possible threats affecting assets and corresponding non-functional properties. A threat is any possible event with undesired consequences [11]. This phase defines a set $\{\langle A_i, p_j, t_k\rangle\}$, where A_i and p_j are identified during phase asset assessment in Section III-A, and t_k is a threat. Each triple $\langle A_i, p_j, t_k\rangle$ forms a *risk*, and is assigned with a score, named *likelihood*, as follows.

Definition 2 (Likelihood): Let A_i be an asset, p_j be a property, and t_k be a threat affecting (A_i, p_j). The likelihood that t_k is exploited on (A_i, p_j) is defined as a function $\mathcal{L}^{t_k}_{(A_i,p_j)}$: $\{A_1,\ldots,A_n\}\times\{p_1,\ldots,p_m\}\times\{t_k,\ldots,t_l\} \to \mathbb{N}$.

The higher the likelihood, the higher the risk associated with (A_i, p_j). Typically, threat assessment considers different factors including the threat context and the scoring systems (e.g., CVSS [15]) for software vulnerabilities [16]. Similarly to asset assessment, we do not restrict likelihood calculation to any particular methods. We note that this phase may require to refine asset assessment in Section III-A, if new assets are discovered.

Example 2: Following Example 1, two threats affect A_{ht}: *i) passive interference* t_{inf} due to environmental noise and *ii) active jamming* t_{jam} due to intentional attacks. According to our reference scenario and assuming the likelihood to take values in $[1, 5]$, a medium likelihood is estimated: $\mathcal{L}^{t_{\text{inf}}}_{(A_{ht},p_I)}$=3, $\mathcal{L}^{t_{\text{jam}}}_{(A_{ht},p_I)}$=3, $\mathcal{L}^{t_{\text{inf}}}_{(A_{ht},p_A)}$=3 and $\mathcal{L}^{t_{\text{jam}}}_{(A_{ht},p_A)}$=3.

Each risk triple is then assigned with a score measuring its qualitative risk value, as follows.

Definition 3 (Risk Value): Let $\langle A_i,\ p_j,\ t_k\rangle$ be a risk. The qualitative risk value is defined as a function $\mathcal{R}^{t_k}_{(A_i,p_j)} = \mathcal{I}_{(A_i,p_j)} \times \mathcal{L}^{t_k}_{(A_i,p_j)}$.

In the following, when clear from the context, we call risk either a risk $\langle A_i,\ p_j,\ t_k\rangle$ or a risk value $\mathcal{R}^{t_k}_{(A_i,p_j)}$.

The total risk of an asset A under property p is then defined according to the following equation.

$$\mathcal{R}_{(A,p)} = \sum_{\langle A,p_j,t_k\rangle \in A.\{r\} | p=p_j} \mathcal{R}^{t_k}_{(A,p_j)}, \tag{1}$$

where $A.\{r\}$ indicates all the risks insisting on asset A. Similarly, the total risk under a property p_j, denoted as $\mathcal{R}_{p_j}$, can be calculated by summing up the total risk of each asset under such a property. Finally, the overall total risk can be calculated by summing up the total risk under each property. We note that operators $\times$ in Definition 3 and $\sum$ in equation 1 follow common practices and can be replaced by more sophisticated functions.

Example 3: Following Example 2, two risks affecting asset A_{ht} under property p_A are $\langle A_{ht}, p_A, t_{\text{inf}}\rangle$ with $\mathcal{R}^{t_{\text{inf}}}_{(A_{ht},p_A)}$=3×4=12 and $\langle A_{ht}, p_A, t_{\text{jam}}\rangle$ with $\mathcal{R}^{t_{\text{jam}}}_{(A_{ht},p_A)}$=3×4=12. We note that the whole set of risks is discussed in Section VI-B.

C. Risk Treatment

Phase risk treatment defines a plan to cope with the identified risk in Section 3. It can include *risk avoidance* (e.g., asset removal), *risk mitigation* (e.g., mechanism deployment), and *risk transfer* (e.g., risk insurance) [17]. Here, we focus on *risk mitigation* that computes the residual risk according to the identified risk and applied security mechanisms.

Phase risk treatment first specifies a mapping between assets A_i, properties p_j, and mechanisms m_l operating on them. In particular, a function θ: $(A_i, p_j) \to \{m_l\}$ retrieves the set of mechanisms m_l mitigating the risk on pair (A_i, p_j). Phase risk treatment then proposes a mitigation degree as the degree to which m_l mitigates a risk on A_i according to the property p_j it supports. It is formally defined as follows.

Definition 4: Let m_l be a mechanism and p_j be a property. The mitigation degree is defined as a function $\delta_{p_j}: m_l \to [0, 1]$.

We note that a mechanism m_l supports a property p_j iff $\delta_{p_j}(m_l) > 0$. We also note that our approach departs from the assumption that risk mitigation is an all-or-nothing approach, where the risk is fully mitigated if the mechanism works as expected; it is not mitigated at all, otherwise.

Example 4: Following Example 3, mechanisms {Anti-Jamming, Physical Protection} mitigate the risk on (A_{ht}, p_A). The mitigation degrees are set to δ_{p_A}(Anti-Jamming)=0.8 and δ_{p_A}(Physical Protection)=0.8. The whole set of mechanisms is discussed in Section VI-B.

IV. Flow Network-based Risk Modeling

The problem of finding how much risk an organization can mitigate can be modeled and solved as a maximum flow problem on a flow network. A flow network is a directed graph where *i)* arcs have capacities indicating the upper bound to the flow that can pass on them; *ii)* a flow starts from a source node s and reaches a sink node t with no *dispersion* [18]. We define a flow network $\mathcal{G}_{p_j}$ for each property p_j in Section III-A, to model the risk mitigation introduced by mechanisms m_l on each pair (A_i, p_j), as follows.

Definition 5: The flow network for property p_j is a triple of the form $\mathcal{G}_{p_j}=\langle V, E, c\rangle$ where: $V = \{A_i\}\cup\{m_l\}\cup\{s, t\}$ is the set of nodes, including assets A_i, mechanisms m_l, source node s, and sink node t; $E = \{(i, l) \mid i \in \{A_i\} \wedge l \in \{m_l\} \wedge m_l \in \theta(A_i, p_j), \forall A_i \in V\} \cup \{(s, A_i), \forall A_i \in V\} \cup \{(m_l, t), \forall m_l \in V\}$ is the set of arcs; c is an *arc capacity function* of the form c: $E \to \mathbb{R}^+$, defined as:

$$c(x,z)\begin{cases} \mathcal{R}_{(A_z,p_j)} & x = s \wedge z \in \{A_i\} \quad (2a)\\ \delta_{p_j}(m_x) \times \mathcal{R}_{(A_i,p_j)} & x \in \{m_l\} \wedge z = t \quad (2b)\\ \infty & x \in \{A_i\} \wedge z \in \{m_l\} \quad (2c)\end{cases}$$

We note that this graph follows the standard practices for building a flow network [19]. It contains a dummy source s and sink t; the remaining nodes form a bipartite graph of assets $\{A_i\}$ and mechanisms $\{m_l\}$ mitigating risk thereon. In case a mechanism operates on more assets, the corresponding mechanism node must be repeated for each asset it operates on. The arcs connecting assets and mechanisms are drawn according to mapping θ in Section III-C and the current set of mechanisms implemented by the organization under assessment. Each asset node is also connected with source s, while each mechanism node with sink t. Arc capacities are

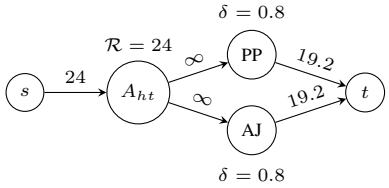

Figure 2. Excerpt of $\mathcal{G}_{p_A}$. We report the total risk of the asset under p_A as $\mathcal{R}$, and the mitigation degree of each mechanism under p_A as δ. Mechanism names have been abbreviated using acronyms.

finally set according to assets' total risk and mechanisms' risk mitigation degree modeling the risk as the flow passing in the network. The total risk $\mathcal{R}_{(A_z,p_j)}$ associated with each asset A_z is set as the capacity of the arc connecting s with A_z, as defined in equation 2a. The risk mitigation degree $\delta_{p_j}(m_x)$ associated with each mechanism m_x, multiplied by asset's total risk $\mathcal{R}_{(A_z,p_j)}$ that m_x mitigates, is set as the capacity of the arc connecting m_x with t, as defined in equation 2b. Capacities on the remaining arcs are set to ∞, as defined in equation 2c, since such arcs encode only the mapping between assets and mechanisms, while capacities are not relevant.

Example 5: Following Example 4, Figure 2 shows the graph corresponding to property p_A. It contains only asset A_{ht} and mechanisms Anti-Jamming and Physical Protection, abbreviated as AJ and PP, respectively. The capacity on arc (s, A_{ht}) is set to 24, corresponding to the total risk of A_{ht} (equation 2a); the capacity on arcs (PP, t) and (AJ, t) is to set to 19.2, corresponding to the total risk of A_{ht} multiplied by the mitigation degree of such mechanisms (equation 2b); the remaining capacities are set to ∞ (equation 2c).

According to the flow network, the risk originating from s must reach t without violating capacities [19]. Hence, the maximum flow f on $\mathcal{G}_{p_j}$ corresponds to the maximum risk under property p_j the organization can mitigate with the mechanisms it has implemented. It can be calculated either per-asset or per-property across all assets. Following equation 1 indicating the total risk of an asset under a property, the total mitigated risk of an asset A_i under a property p_j is:

$$\sum_{m_l \in \theta(A_i,p_j)} f(A_i, m_l), \tag{3}$$

where $f(i, l)$ indicates the amount of flow passing on arc (i, l). In other words, it corresponds to the amount of risk actually flowing out of asset node A_i. From equation 3, we can derive the *residual risk* for an asset under a property as:

$$\mathcal{R}_{(A_i,p)} - \sum_{m_l \in \theta(A_i,p_j)} f(A_i, m_l), \tag{4}$$

where the minuendo corresponds to equation 3.

We note that the overall residual risk can be easily computed per-property as the difference of the sum over the total risk of all assets and the maximum flow on the graph. Also, if all arcs (s, A_i) are saturated, then all assets' total risk have been mitigated, that is, no residual risk remains. Moreover, the total amount of mitigated (residual, resp.) risk under all properties can be computed by summing up the mitigated (residual, resp.) risk under each property.

The process in this section can be fine-tuned, by adjusting nodes and arcs to reflect specific use cases and alternative approaches to *risk mitigation*. For instance, when *risk avoidance* is implemented by removing assets entailing a level of risk, the flow network can be adapted by removing the corresponding asset nodes.

V. Assurance-Based Risk Mitigation

Finding how much risk the organization can mitigate is the very first outcome of a risk management framework. However, this activity relies on the strong assumption that mechanisms work as expected and mitigate the risk on the corresponding assets at their full potential. This assumption can lead to scenarios where the residual risk is underestimated, possibly resulting in catastrophic consequences. We relax this assumption by coupling risk management with assurance techniques to dynamically adjust the risk mitigation degrees, and, ultimately, the residual risk. Assurance techniques aim to verify the support for a non-functional property, and contribute to consolidate and validate residual risk calculation [3], using a two-steps process consisting of *i) assurance evaluation*; *ii) residual risk calculation*. Assurance evaluation adjusts the risk mitigation degree of a given mechanism in Definition 4 on the basis of the actual (set of) non-functional property it supports. Residual risk calculation refines the residual risk in equation 4 on the basis of the assurance evaluation outcome, in general, and the supported property, in particular.

A. Assurance Evaluation

An assurance evaluation is a process that receives as input a set of mechanisms and produces as output the set of properties proven on the them, if any. The property is the means to confirm whether the mechanism works and in turn mitigates the risk on the asset as expected (see Section III-C).

Our approach based on assurance evaluation departs from existing solutions where flat classes of properties are used (such as confidentiality, integrity, and availability in Section III-A), introducing an approach where properties are distributed in a hierarchy [20]. Each mechanism can in fact support a class of properties at different levels of strength, thus affecting the degree to which it really mitigates the risk. To this aim, each property p_j is first refined with a set of attributes, forming a pair $(\widehat{p}_j, Attr)$, where $\widehat{p}_j$ is the property name corresponding to a class of properties and $Attr$ is a set of pairs (a_i, v_i) with a_i the attribute name and v_i the corresponding value. Attributes take value in a specific domain over which a total order relationship $\geq_{a_i}$ is defined, s.t. $(a_i, v_i) \geq_{a_i} (a_i, v_j)$ iff v_i is *stronger* than v_j. A hierarchy of properties is then defined as follows.

Definition 6 (Property Hierarchy): A hierarchy of properties is a pair $(\mathcal{P}, \succeq_p)$ where $\mathcal{P}$ is a set of properties $(\widehat{p}_j, Attr)$, and $\succeq_p$ is partial order relationship over $\mathcal{P}$ s.t. $\forall p_x, p_z \in \mathcal{P}, p_x \succeq_p p_z \iff p_x.\widehat{p} = p_z.\widehat{p} \wedge \forall (a_k, v_k) \in Attr, p_x.(a_k, v_k) \geq_{a_k} p_z.(a_k, v_k)$.

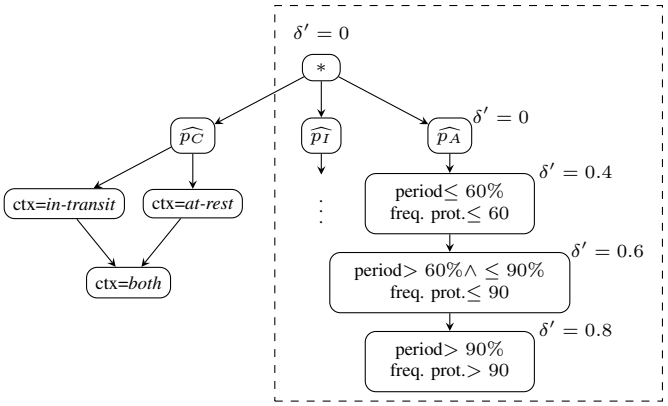

Figure 3. Excerpt of a hierarchy of properties.

We note that $(*, \emptyset)$ is the root of hierarchy $(\mathcal{P}, \succeq_p)$. Each mechanism m_l induces a view over $(\mathcal{P}, \succeq_p)$, such that:

i) $(*, \emptyset)$ is the root of the view;
ii) for each class of properties $\widehat{p}_j$ supported by m_l, it exists a subtree routed in $(\widehat{p}_j, \emptyset)$;
iii) for each $(\widehat{p}_j, \emptyset)$, all properties $(\widehat{p}_j, Attr)$ relevant for m_l are kept in the view.

Each element p_j of the view is annotated with the actual risk mitigation degree of m_l under p_j, denoted as δ'_{p_j}. Intuitively, annotations follow the tree-like structure of the hierarchy, with highest values in the leaves and δ'_{root}=0. In other words, a view induced by a mechanism contains all properties it can support, where annotations indicate the actual mitigation degrees a mechanism supporting a specific property guarantees.

Example 6: Following Example 5, Figure 3 shows an excerpt of a hierarchy of properties. The view induced by mechanism Anti-Jamming is represented by the dashed box, consisting of the subtrees rooted at properties p_I and p_A. In particular, attribute *period* indicates the expected uptime guaranteed by the mechanism, and *frequency protection*, abbreviated as *freq. prot.*, indicates the percentage of the frequency spectrum where the devices operate and covered by the mechanism. Each property in the hierarchy is associated with a mitigation degree. For instance, the mitigation degree of mechanism Anti-Jamming under property p_1=($\widehat{p}_A$,{(period, 95%),(freq. prot., 95)}) is δ_{p_1}(Anti-Jamming)$'$=0.8.

The assurance process is driven by an *assurance model* that specifies all the activities to be executed to prove a property p_j on a given mechanism m_l. When p_j is proven for m_l, its mitigation degree is adjusted to reflect the observed behavior. An assurance model is formally defined as follows.

Definition 7 (Assurance Model): An assurance model is a triple $\langle m_l, p_j, \{tc_i\}\rangle$, where m_l is the mechanism target of the evaluation, p_j is the property whose support must be proven on m_l, and $\{tc_i\}$ is the set of test cases used to verify the support of p_j by m_l.

A property is successfully evaluated for a given mechanism iff the evidence collected using the test cases supports it. We note that three possible assurance evaluations can be executed depending on the scenario, as discussed below.

C1) m_l has been already proven to support p_j. No activities are needed and a risk mitigation degree is applied according to p_j;
C2) m_l is released with a target property p_j. A single assurance evaluation is executed to prove support for p_j. The mechanism's mitigation degree $\delta_{p_j}(m_l)$ is adjusted with the annotation corresponding to p_j in the hierarchy, that is, $\delta_{p_j}(m_l)'$; $\delta_{p_j}(m_l)'$=0, if it is not supported.
C3) No knowledge about m_l and p_j is available. An assurance evaluation is executed for each property in the view induced by m_l on hierarchy $(\mathcal{P}, \succeq_p)$. The adjusted mechanism's mitigation degree $\delta_{p_j}(m_l)'$ is the one of the best supported property p_j in the hierarchy.

We note that *C1)* is a simplification of *C2)* as a property has been already evaluated, while *C3)* is a generalization of *C2)* since, in the worst case, an assurance evaluation can be executed for any possible properties. In the following, for simplicity but no lack of generality, we focus on *C2)* that is the approach currently used in the assurance literature [20].

B. Assurance-Based Residual Risk

The final step of our risk management framework calculates the assurance-based residual risk. Upon executing the assurance evaluation in Section V-A for all mechanisms in our flow networks, a set of adjusted risk mitigation degrees δ'_{p_j} is retrieved. New flow networks $\mathcal{G}'_{p_j}$ are built as in Definition 5. The only difference is on the capacity function c of arcs (m_l, t), which is set according to the adjusted mitigation degrees, as

$$\delta_{p_j}(m_l)' \times \mathcal{R}_{(A_i, p_j)} \tag{5}$$

The maximum flow on these graphs correspond to the maximum amount of risk the organization actually mitigates, per-asset, per-property, and globally (Section IV).

VI. Experiments

We experimentally evaluated the performance and quality of the proposed approach in a simulated environment. Experiments have been run on a laptop equipped with an Intel® Core i7-5500U @ 2.4 GHz (2 cores, 4 threads), 16 GBs of RAM, operating system Ubuntu 20.04 x64, Python runtime 3.8.6.

A. Performance

We evaluated the performance of our framework by measuring the time needed to compute the maximum flow on randomly-generated graphs. We used a popular Python graph library [21] whose maximum flow algorithm runs in $\mathcal{O}(|V|^2\sqrt{|E|})$ [18], [22]. We generated the graphs varying the number of asset nodes in $50, 100, 150, 200, 250, 300$, denoted as $|\{A_i\}|$, and the number of mechanisms operating on each asset in $4, 8, 16$, denoted as $|\theta|$. We randomly chosen the mapping between assets and mechanisms in Section III-C, the total risk of each asset in $[1, 50]$, and the mitigation degree of each mechanism in $[0.1, 1]$. Figure 4 shows that the execution time never exceeded 0.25ms, even when considering relatively big and dense graphs, proving that our approach is feasible

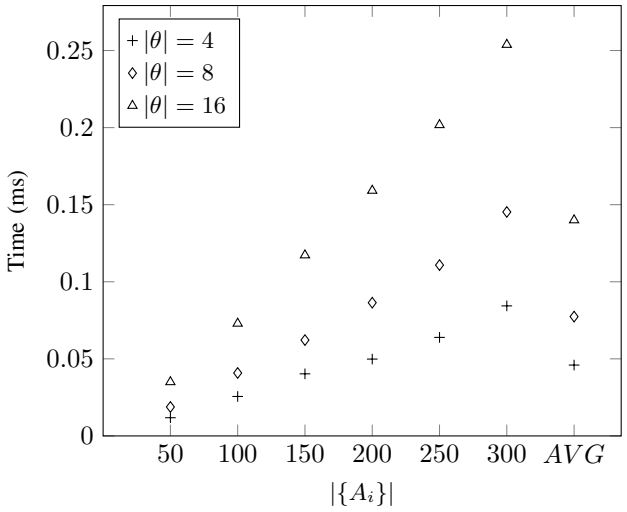

Figure 4. Performance results. Column "AVG" indicates the overall average.

and easily applicable even to very complex scenarios. We note that assurance cost has not been taken into account, as it is considered a *fixed cost* not impacting on the performance of risk calculation.

B. Quality

We evaluated the quality of our framework in a complete walkthrough based on the reference scenario in Section II-B. Table II presents the risks retrieved from phases asset assessment, threat assessment, and risk calculation in Sections III-A and III-B. As an example, asset A_{ht} is mostly affected by threats Interference, Jamming, Spoofing or DoS, impacting on properties p_I and p_A. On the contrary, asset A_p, which is a trade secret, is mostly affected by threats Eavesdropping and Filter bypass, impacting on properties p_C and p_A.

Tables III(a)–III(c) present the mechanisms resulting from phase risk treatment in Section III-C. For brevity, we report in the same table both the expected mitigation degree δ and the adjusted mitigation degree δ' resulting from assurance evaluation. Table III(a) presents the mechanisms implemented to protect assets A_{ht} and A_p with respect to property p_C. They are PINs for RFID authentication, Secure Pairing and Public Keys certificates for Bluetooth pairing. Tables III(b) and III(c) present the mechanisms to protect assets A_{ht} and A_p with respect to properties p_I, p_A. They are Anti-Jamming and Physical Protection to avoid signal disruptions, Code checks at microcontroller-level, Secure Pairing and Public Keys certificates.

We built three graphs, one for each property, to retrieve the mitigated and residual risks according to how mechanisms are expected to work. Then, we built the corresponding graphs after the application of assurance evaluation in Section V-B. Figure 5 shows the flow networks $\mathcal{G}_{p_A}$ and $\mathcal{G}'_{p_A}$ for property p_A, where dashed circles represent those mechanisms with changed mitigation degree. Table IV shows the results of our approach aggregated per property, where "Without" ("With", resp.) denotes the results without assurance evaluation (with assurance evaluation, resp.). When assurance evaluation was not applied, most of the risk was mitigated by the implemented mechanisms; for instance, the risk under property p_I was completely mitigated, that is, residual risk $= 0$. When assurance evaluation was applied to retrieve the specific properties

Table II
RISKS

Property	Asset	$\mathcal{I}$	Threats	$\mathcal{L}$	$\mathcal{R}^t_{(A,p)}$	$\mathcal{R}_{(A,p)}$
p_C	A_{ht}	1	Eavesdrop.	4	4	6
			RFID impers.	2	2	
p_C	A_p	4	Eavesdrop.	4	16	52
			RFID impers.	2	8	
			Pairing attacks	2	8	
			Adv. attacks	2	8	
			Filter bypass	3	12	
p_I	A_{ht}	4	Interfer. (t_{inf})	3	12	52
			Jam. (t_{jam})	3	12	
			Removal	2	8	
			Collision	1	4	
			Spoofing	1	4	
			Sensor errors	3	12	
p_I	A_p	4	Interfer. (t_{inf})	1	4	8
			Jam. (t_{jam})	1	4	
p_A	A_{ht}	4	Interfer. (t_{inf})	3	12	64
			Jam. (t_{jam})	3	12	
			Removal	2	8	
			Collision	1	4	
			Spoofing	1	4	
			Sensor errors	3	12	
			DoS	3	12	
p_A	A_p	4	Interfer. (t_{inf})	1	4	44
			Jam. (t_{jam})	1	4	
			Pairing attacks	2	8	
			Adv. attacks	2	8	
			Filter bypass	3	12	
			DoS	2	8	

Table III
MECHANISMS

Mechanism	p_C		
	Assets	δ	δ'
PINs	A_p	0.9	0.7
Secure Pairing	A_p	0.9	0.4
Public Keys	A_{ht}, A_p	0.9	0.8

(a) Property p_C

Mechanism	p_I		
	Assets	δ	δ'
Anti-Jam.	A_{ht},A_p	0.8	0
Physic. Protect.	A_{ht}	0.8	0.3
Code checks	A_{ht}	0.9	0.5
Secure Pairing	A_p	0.9	0.4
Public Keys	A_{ht}	0.9	0.7

(b) Property p_I

Mechanism	p_A		
	Assets	δ	δ'
Anti-Jam.	A_{ht},A_p	0.8	0
Physic. Protect.	A_{ht}	0.8	0.3
Code checks	A_{ht}	0.9	0.5
Public Keys	A_{ht}	0.9	0.7

(c) Property p_A

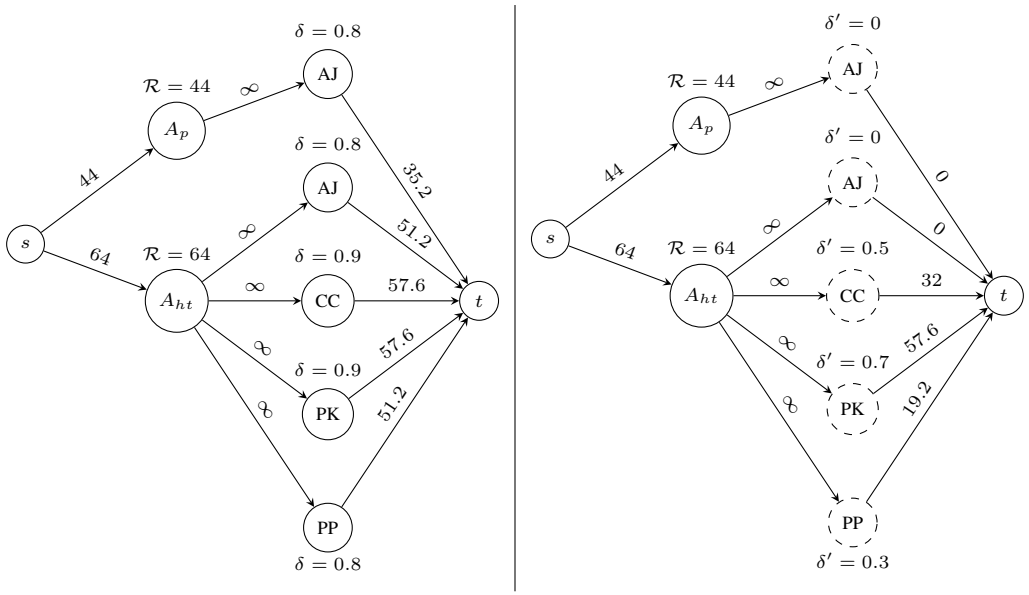

Figure 5. $\mathcal{G}_{p_A}$ and $\mathcal{G}'_{p_A}$ according to the identified risks and mechanisms. Mechanism names have been abbreviated using acronyms, and dashed circles represent mechanisms whose mitigation degree changed after assurance evaluation.

Table IV
RESIDUAL RISKS

Property	$\mathcal{R}_p$	Mitigated Risk		Residual Risk		% diff.
		Without	With	Without	With	
p_C	58	57.4	56.8	0.6	1.2	10.35
p_I	60	60	55.2	0	4.8	8
p_A	108	99.2	64	8.8	44	32.59

supported by the implemented mechanisms the residual risk raised to: *i)* 1.2 under property p_C, with an average increase of 10.35%; *ii)* 4.8 under property p_I, with an average increase of 8%; *iii)* 44 under property p_A, resulting in an increase of 32.59%. Table III shows that most of the mechanisms did not work as expected, as their support for the properties was much lower than expected, resulting in lower mitigation degrees. We observe that, for properties p_C and p_I, mechanisms could mitigate most of the risk reducing fluctuations in residual risk. By contrast, Figure 5 shows that the residual risk under p_A was equal to the total risk of asset A_p, meaning that no mitigation was applied. The reason is that the only mechanism operating on A_p was Anti-Jamming, which, after assurance evaluation, had an adjusted mitigation degree δ'=0.

To conclude, our results stress the importance of having an assurance-based view of the risk. It clearly emerges that high variations between expected and adjusted mitigation degrees denote a high discrepancy between what the organization *thinks to have* and what it *actually has*, in terms of risk mitigation. For instance, using existing solutions, the organization may think that risk on assets A_{ht} and A_p is mitigated by mechanism Anti-Jamming under property p_A, while it is found to be false under our framework. The integration of assurance techniques within a risk management framework permits to assess the behavior of the mechanisms, possibly finding scenarios in which they do not work at their full potential, thus resulting in residual risk higher than expected. Residual risk can then be properly adjusted, increasing organization's consciousness and avoiding dramatic consequences, as in the case of the Capital One data breach [4]. The incident was the result of an unauthorized access to a subset of the company servers hosted at Amazon Web Service (AWS), which took place on March 22 and 23, 2019. The attack exploited the misconfigured Web Application Firewall (WAF) deployed by the company, allowing attackers to access private endpoints. The attack has been discovered only after 3 months and affected 106 million customers, resulting in 15% loss of the company shares within the following two weeks.

VII. RELATED WORK

Risk management and assurance evaluation have been considered in several works in the area of distributed system security and assessment. Different assurance and risk management frameworks have been defined, which are discussed in the following of this section.

Assurance Frameworks. Assurance frameworks have recently received increasing attention as the means to gain justifiable confidence that IT systems will consistently demonstrate one or more security (non-functional in a broader sense) properties, and behave as expected [20], [23]. Anisetti et al. [20] presented an assurance framework for test-based cloud certification, which has then been extended for hybrid systems mixing cloud and private infrastructures [24]. Aslam et al. [25] presented *FoNAC*, an audit and certification framework targeting fog computing and based on Trusted Platform Modules. De la Vara et al. [26] introduced a certification framework targeting cyber-physical systems out of the European research

project AMASS, for which they provide several tools aiding the certification process.

Monitoring-based assurance is also a flourishing research line, for which Aceto et al. [27] provided a thorough survey. Povedano-Molina et al. [28] discussed *DARGOS*, a highly scalable cloud monitoring solution. Alcaraz Calero et al. [29] and De Chaves et al. [30] presented two frameworks based on the monitoring tool *NAGIOS*.

Security assurance also provides a plethora of techniques to consolidate user confidence in IT systems. In this paper, we rely on test-based techniques to adjust residual risk, an approach that, to the best of our knowledge, has never been applied in the context of risk management frameworks.

Risk Management Frameworks. Research on risk management is a consolidated area that dates back to the 90s [17], [31]. Many risk management frameworks have been presented through the years, though they do not address the intrinsic complexity and heterogeneity of modern systems. Existing frameworks (e.g., [7], [8]) mostly focus on static environments and monolithic systems limiting their practical applicability to modern scenarios. A comprehensive and novel taxonomy about risk frameworks has been provided in [17], where the authors claim that the traditional *qualitative vs quantitative* classification is not sufficiently expressive. In general, quantitative methodologies, which rely on probability and statistics to quantify risk, are appealing but considered lengthy and difficult. Here, we classify risk management frameworks according to the peculiarities of the approach and the respective target: *i) traditional frameworks*, *ii) IoT-based frameworks*, *iii) graph-based frameworks*, *iv) machine learning-based frameworks*.

Traditional frameworks refer to traditional standards and generic approaches. Among them, ISO 31000 [7], NIST SP800-30 [8] and OCTAVE [13] stand out. They provide a set of guidelines and steps to follow to identify and properly treat risks. In particular, NIST SP800 proposes to start by identifying threats rather than assets, as opposite to OCTAVE. The integration of risk and testing has then been outlined in a framework proposed by ETSI [3], where two approaches have been proposed: testing-driven risk assessment (like the one in this paper) and risk-driven security testing. Researchers presented several risk frameworks based on many different approaches [17]. In general, fuzzy theory is often used, as it fits well the intrinsic uncertainty of risk (e.g., Guan et al. [32], Deng et al. [33]).

IoT-based frameworks try to address the peculiarities of IoT. Nurse et al. [1] identified the main gaps in such scenario, including the need for continuous assessment, which should cope with high dynamism and limited system knowledge. Radanliev et al. [34] provided another survey, where identified gaps, standards, and best practices are used to introduce a conceptual risk framework for IoT. Similarly, Kandasamy et al. [35] introduced a qualitative risk framework for Internet of Medical Things (IoMT) based on hard-coded values. Matheu-García et al. [16] proposed a certification-based methodology targeting IoT devices. It builds on the ETSI proposal [3] by integrating risk during the certification process, and includes a visual multidimensional representation of the results.

Graph-based frameworks use graphs and graph-like structures to aid (a subset of) risk management phases. Traditionally, graphs have been mostly used for attack and defense modeling, as surveyed by Kordy et al. [36]. A subset of such techniques, *fault* and *attack tree*, has been thoroughly analyzed by Nagaraju et al. [31]. A fault tree identifies all the events that could contribute to a failure, along with the respective frequencies; an attack tree identifies the possible ways to carry out an attack. Recently, graphs have also been used for other objectives. Sawilla et al. [37] presented a framework aimed to rank the importance of assets w.r.t. attacks using a graph model and a custom algorithm named *AssetRank*. Alpcan et al. [38] introduced a framework aiming at propagation, capturing the complex interactions among model's objects. Zhu et al. [39] exploited graphs to address attacks' impacts against critical infrastructures. Their work can express the impact on individual nodes and the respective propagation towards the other nodes in the graph. Jahnke et al. [6] used graphs to quantify the effects of attacks and corresponding countermeasures. The same aspect has been analyzed by Kheir et al. [40] with a different dependency-based model.

Machine-learning based frameworks use machine learning within risk frameworks. Hegde et al. [41] presented a survey encompassing several industry areas. Among the most flourishing areas, there are automotive, construction and railways, while cyber security is at the opposite, being one of the least flourishing. Bilge et al. [42] proposed *RiskTeller*, a solution predicting which machines are at high risk of infections with months in advance. It is based both on supervised and semi-supervised machine learning. Sun et al. [43] thoroughly analyzed the trend of incidents prediction.

To conclude, none of the existing risk management solutions even come close to tackle all the requirements in Section II-A. Rather, they often take a vertical approach, addressing some specific issues. One of the most relevant gaps, highlighted in many works, is the need for a continuous process, especially considering the high dynamism of emerging systems. At the same time, propagation is often understood as another key point. Building on this, abstraction, scalability and automation are fundamental as well. Finally, the integration with assurance techniques is often not stressed enough, culminating in results which are far from reality, leading to a wrong decision-making. The framework in this paper takes a horizontal approach to get a qualitative and realistic picture of risk, using an intuitive model, as discussed in Section VI-B.

VIII. Conclusions

The ability to identify, evaluate and manage risks affecting an organization is a fundamental aspect supporting ICT evolution. This is especially critical in modern distributed systems, where the convergence between cloud computing and IoT results in complex and dynamic systems. Existing risk practices must then be rethought and redesigned towards a continuous and adaptive approach.

In this paper, we presented a risk framework for modern distributed systems which adheres to well-known qualitative approaches, centered around the concept of non-functional properties driving the overall process. The framework is fully integrated with assurance techniques, enabling to adjust risk on the basis of assurance evaluation activities. Also, it is based on an intuitive model that maps risk to flow on flow networks. We believe approaches of this type are going to be fundamental in modern distributed environments, leading to the construction of systems allowing users to enjoy the benefits of emerging technologies, while giving to enterprises awareness and control over their risks.

The paper leaves space for future work. First, propagation of threats/attacks in Section II-A will be managed to fully unleash the potential of our framework. Second, a cost model will be integrated in our framework to retrieve the optimum solution that balances residual risk and costs. Third, a continuous risk management solution will be provided to support an adaptive approach to risk minimization.

ACKNOWLEDGMENTS

Research supported, in parts, by EC H2020 Project CONCORDIA GA 830927 and Università degli Studi di Milano under the program "Piano sostegno alla ricerca".

REFERENCES

[1] J. R. C. Nurse, S. Creese, and D. De Roure, "Security Risk Assessment in Internet of Things Systems," *IT Professional*, vol. 19, no. 5, 2017.

[2] P. Radanliev, D. Charles De Roure, J. Nurse, P. Burnap, E. Anthi, A. Uchenna, L. Maddox, O. Santos, and R. Mantilla Montalvo, "Cyber risk management for the internet of things," *Preprints 2019*, 2021.

[3] "Methods for Testing & Specification; Risk-based Security Assessment and Testing Methodologies," European Telecommunications Standards Institute, Sophia Antipolis Cedex, France, Standard, January 2016.

[4] N. N. Neto, S. E. Madnick, A. M. G. de Paula, and N. M. Borges, "A Case Study of the Capital One Data Breach," *SSRN Electronic Journal*, January 2020.

[5] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges," in *Proc. of FIT 2012*, Islamabad, Pakistan, December 2012.

[6] M. Jahnke, C. Thul, and P. Martini, "Graph based Metrics for Intrusion Response Measures in Computer Networks," in *Proc. of IEEE LCN 2007*, Dublin, Ireland, October 2007.

[7] "Risk management – Guidelines," International Organization for Standardization, Geneva, CH, Standard, February 2018.

[8] Joint Task Force Transformation Initiative, "Guide for Conducting Risk Assessments," National Institute of Standards and Technology, Gaithersburg, MD, Tech. Rep. NIST Special Publication (SP) 800-30, Rev. 1, September 2012.

[9] Miao Wu, Ting-Jie Lu, Fei-Yang Ling, Jing Sun, and Hui-Ying Du, "Research on the architecture of Internet of Things," in *Proc. of ICACTE 2010*, Chengdu, China, August 2010.

[10] Zhihong Yang, Yingzhao Yue, Yu Yang, Yufeng Peng, Xiaobo Wang, and Wenji Liu, "Study and application on the architecture and key technologies for IOT," in *Proc. of ICMT 2011*, Hangzhou, China, July 2011.

[11] D. J. Landoll, *The Security Risk Assessment Handbook: A Complete Guide for Performing Security Risk Assessments*. Auerbach Publications, 2005.

[12] "Information technology – Security techniques – Information security management systems – Requirements," International Organization for Standardization, Geneva, CH, Standard, October 2013.

[13] B. Tucker, "Advancing Risk Management Capability Using the OCTAVE FORTE Process," Carnegie Mellon University, Tech. Rep., 2020.

[14] "Security and resilience – Business continuity management systems - Requirements," International Organization for Standardization, Geneva, CH, Standard, October 2019.

[15] "Common vulnerability scoring system version 3.1," Forum of Incident Response and Security Teams, Standard, June 2019.

[16] S. N. Matheu-García, J. L. Hernández-Ramos, A. F. Skarmeta, and G. Baldini, "Risk-based automated assessment and testing for the cybersecurity certification and labelling of IoT devices," *Computer Standards & Interfaces*, vol. 62, February 2019.

[17] A. Shameli-Sendi, R. Aghababaei-Barzegar, and M. Cheriet, "Taxonomy of Information Security Risk Assessment (ISRA)," *Computers & Security*, vol. 57, March 2016.

[18] A. V. Goldberg and R. E. Tarjan, "A New Approach to the Maximum-Flow Problem," *Journal of the ACM*, vol. 35, no. 4, October 1988.

[19] J. Kleinberg and E. Tardos, *Algorithm design*. Pearson Education India, 2006.

[20] M. Anisetti, C. A. Ardagna, E. Damiani, and F. Gaudenzi, "A Semi-Automatic and Trustworthy Scheme for Continuous Cloud Service Certification," *IEEE Transactions on Services Computing (TSC)*, vol. 13, no. 1, 2020.

[21] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring Network Structure, Dynamics, and Function using NetworkX," in *Proc. of SciPy 2008*, Pasadena, CA, USA, August 2008.

[22] L. Tunçel, "On the complexity of preflow-push algorithms for maximum-flow problems," *Algorithmica*, vol. 11, no. 4, April 1994.

[23] C. Ardagna, R. Asal, E. Damiani, and Q. Vu, "From security to assurance in the cloud: A survey," *ACM Computing Surveys (CSUR)*, vol. 48, no. 1, August 2015.

[24] M. Anisetti, C. A. Ardagna, N. Bena, and E. Damiani, "Stay Thrifty, Stay Secure: A VPN-Based Assurance Framework for Hybrid Systems," in *Proc. of SECRYPT 2020*, Paris, France, July 2020.

[25] M. Aslam, B. Mohsin, A. Nasir, and S. Raza, "FoNAC - An automated Fog Node Audit and Certification scheme," *Computers & Security*, vol. 93, June 2020.

[26] J. L. de la Vara, A. Ruiz, B. Gallina, G. Blondelle, E. Alaña, J. Herrero, F. Warg, M. Skoglund, and R. Bramberger, "The AMASS Approach for Assurance and Certification of Critical Systems," in *Proc. of Embedded World 2019*, Norimberg, Germany, February 2019.

[27] G. Aceto, A. Botta, W. de Donato, and A. Pescapè, "Cloud monitoring: A survey," *Computer Networks*, vol. 57, no. 9, June 2013.

[28] J. Povedano-Molina, J. M. Lopez-Vega, J. M. Lopez-Soler, A. Corradi, and L. Foschini, "DARGOS: A highly adaptable and scalable monitoring architecture for multi-tenant Clouds," *Future Generation Computer Systems*, vol. 29, no. 8, October 2013.

[29] J. M. Alcaraz Calero and J. G. Aguado, "MonPaaS: An Adaptive Monitoring Platformas a Service for Cloud Computing Infrastructures and Services," *IEEE Transactions on Services Computing (TSC)*, vol. 8, no. 1, 2015.

[30] S. A. De Chaves, R. B. Uriarte, and C. B. Westphall, "Toward an architecture for monitoring private clouds," *IEEE Communications Magazine*, vol. 49, no. 12, 2011.

[31] V. Nagaraju, L. Fiondella, and T. Wandji, "A Survey of Fault and Attack Tree Modeling and Analysis for Cyber Risk Management," in *Proc. of IEEE HST 2017*, Waltham, MA, USA, April 2017.

[32] B.-C. Guan, C.-C. Lo, P. Wang, and J.-S. Hwang, "Evaluation of information security related risks of an organization: the application of the multicriteria decision-making method," in *Proc. of IEEE ICCST 2003*, Taipei, Taiwan, October 2003.

[33] Y. Deng, R. Sadiq, W. Jiang, and S. Tesfamariam, "Risk analysis in a linguistic environment: A fuzzy evidential reasoning-based approach," *Expert Systems with Applications*, vol. 38, no. 12, 2011.

[34] P. Radanliev, D. C. De Roure, J. R. C. Nurse, R. Mantilla Montalvo, S. Cannady, O. Santos, L. Maddox, P. Burnap, and C. Maple, "Future developments in standardisation of cyber risk in the Internet of Things (IoT)," *SN Applied Sciences*, vol. 2, no. 2, January 2020.

[35] K. Kandasamy, S. Srinivas, K. Achuthan, and V. P. Rangan, "Iot cyber risk: a holistic analysis of cyber risk assessment frameworks, risk vectors, and risk ranking process," *EURASIP Journal on Information Security*, vol. 2020, no. 1, May 2020.

[36] B. Kordy, L. Pièvre-Cambacédès, and P. Schweitzer, "Dag-based attack and defense modeling: Dont miss the forest for the attack trees," *Computer Science Review*, vol. 13-14, November 2014.

[37] R. E. Sawilla and X. Ou, "Identifying Critical Attack Assets in Dependency Attack Graphs," in *Proc. of ESORICS 2008*, Málaga, Spain, October 2008.

[38] T. Alpcan and N. Bambos, "Modeling Dependencies in Security Risk Management," in *Proc. of CRiSIS 2009*, Toulouse, France, October 2009.

[39] Q. Zhu, Y. Qin, C. Zhou, and L. Fei, "Hierarchical Flow Model-Based Impact Assessment of Cyberattacks for Critical Infrastructures," *IEEE Systems Journal*, vol. 13, no. 4, December 2019.

[40] N. Kheir, N. Cuppens-Boulahia, F. Cuppens, and H. Debar, "A Service Dependency Model for Cost-Sensitive Intrusion Response," in *Proc. of ESORICS 2010*, Athens, Greece, September 2010.

[41] J. Hegde and B. Rokseth, "Applications of machine learning methods for engineering risk assessment – A review," *Safety Science*, vol. 122, February 2020.

[42] L. Bilge, Y. Han, and M. Dell'Amico, "RiskTeller: Predicting the Risk of Cyber Incidents," in *Proc. ACM SIGSAC CCS 2017*, Dallas, TX, USA, October 2017.

[43] N. Sun, J. Zhang, P. Rimba, S. Gao, L. Y. Zhang, and Y. Xiang, "Data-Driven Cybersecurity Incident Prediction: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, 2019.