Chapter 2

Preliminaries and Literature Review

In this chapter, first, we provide some preliminaries related to the presented work. Then, we perform an in-detailed literature survey on modeling, analysis and organization methods for CPS security to identify issues and research gaps related to the existing works.

2.1 Preliminaries

2.1.1 System Requirements

There are two types of system requirements: (1) functional requirements, and (2) non-functional requirements. Functional requirements are the requirements that are identified to deliver the specified functionalities. Non-functional concerns are the requirements related to quality of services that can be implemented as constraints on functionalities. The non-functional requirements include security, safety, reliability, availability, performance, fault-tolerance and maintainability *etc.*

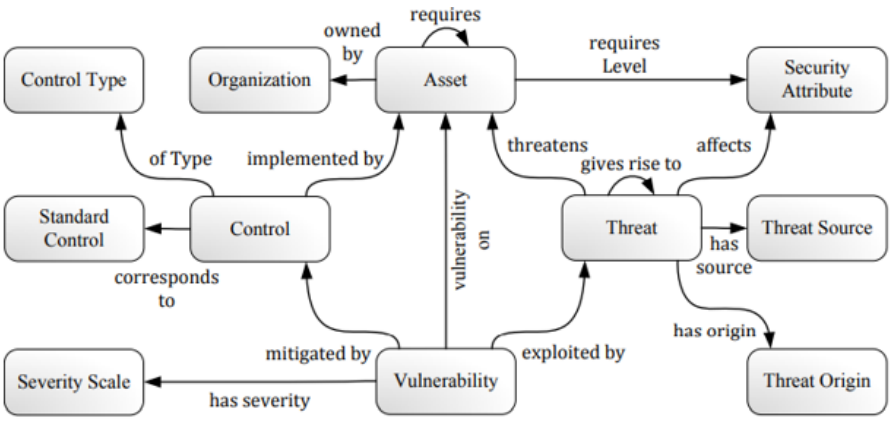

FIGURE 2.1: Security ontology

2.1.2 Security Related Concepts

There are many organizations including National Institute of Standards and Technology (NIST), Industrial Control Systems Computer Emergency Response Team (ICS-CERT) and International Society of Automation/ International Electrotechnical Commission Standards (ISA/IEC-62443) that work for security standards and define security related concepts and their relationships as security ontology [129], shown in FIGURE 2.1. To model and analyse the CPS security, these concepts are important to understand. The description includes:

Asset: A valuable tangible or intangible resource

Vulnerability: A weakness of asset that may be exploited by one or more threats.

Threat: refers to the deliberate faults that leads to possible violation of security criteria.

Threat agent: The agent (program or person) who brings off the threat.

Threat surface: The total number of points where the vulnerabilities can be exploited to gain access and extract the data from the system and environment.

Severity: It is the level of risk such as high, medium, low, and very low *etc.*

Risk: Expected loss to one or more assets due to threat.

Control: A way to enable security requirements to secure the assets.

Security criteria: These are security constraints on assets.

Security requirements: The constraints required to be fulfilled to achieve security goal and mitigate risk.

Security attributes: Security attributes including confidentiality, integrity, availability, non-repudiation, survivability and traceability. Confidentiality refers to the absence of unauthorized information disclosure. Integrity is the absence of unauthorized system state alterations. Availability defines of time or in steady state the readiness of correct service on demand. Non-repudiation is a property that averts false denial of involvement in future by either party in a transaction. Survivability is the system capability to complete its mission within specified time, even in the presence of intentional or unintentional threats.

2.1.3 Model Based System Engineering (MBSE)

MBSE is a formalized methedology that facilitates the process of requirements, design, analysis, verification, and validation associated with the complex system development.

Model: A model is a graphical, mathematical or physical representation of something that abstracts reality to eliminate some complexity. The models show the stakeholders that the presented design satisfies the system's requirements. It is possible to generate partial or complete system implementation from system model. For security analysis, the models should describe when and how security breaches materialize and trace their impact on the system. Moreover, these may apply the defense mechanisms and analyse its effects with respect to defense cost; provide solutions for system recovery, and system maintenance.

Qualitative analysis: The analysis shows the presence or absence of specified properties instead of the estimated range.

Quantitative analysis: The analysis shows measurement of quantities of the particular properties using complex statistical or mathematical modeling.

Modeling language: Modeling language is a terminology for communicating an abstract idea that a model captures. The modeling languages can be formal, semi-formal or informal.

2.1.4 Threat modeling

It is a process to identify, enumerate and mitigate the potential threats. It's main purpose is to provide a systematic analysis of system nature, probable attacker's profile, the most probable attack vectors, the most desired assets by an attacker, and the controls required to defend the system. There are several threat models proposed to identify threats such as STRIDE (Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege) [64], Process for Attack Simulation and Threat Analysis (PASTA) [141], Visual, Agile and Simple Threat(VAST) [124], and Trike [117]. The popular threat modeling language and methods include Unified Modeling Language(UML), combinatorial and state-space based models. The combinatorial models are generally the decision trees that can narrate sets of events that can lead to system failure in a combinatorial way. These models include attack trees, attack defense trees, Bayesian networks *etc.*

2.1.4.1 Markov Model

The basis of Markov models is Markov Chain. It is named after the Russian mathematician Andrey Markov. A Markov chain is a stochastic model which represents a sequence of probable events in which the probability of the next state depends only on the previous event [41].

A stochastic process $\{X(t)|t \in T\}$ is called a Markov process if for any $t_0 < t_1 < .....t_n < t_{n+1}$, the conditional distribution of $X(t_{n+1})$ for given values of $X(t_0), X(t_1), ..., X(t_n)$ depends only on $X(t_n)$ and not on the previous values. The

values that $X(t)$ can assume are in general called 'states', all of which together form a 'state space' Ω.

Markov models are demonstrated as a graphical representation of these chains based on the Markov chain. These models can be seen as working in conjuncture with State machines where the transitions are placed between different system states.

Extending the State machines, Markov models can model and analyse the security of entire control systems by incorporating the vulnerable points. It is done by assigning the probabilities to the transitions, namely the Markov chain probabilities. These transition probabilities are given in the transition matrix P, such as

$$P = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1m} \\ P_{21} & P_{22} & \cdots & P_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ P_{m1} & P_{m2} & \cdots & P_{mm} \end{bmatrix}$$

where,

P_{ij} is the transition probability between state i and j.

Markov processes are categorized into four types based on whether associated time (T) and state space (Ω) are discrete (countable) or continuous (uncountable). Here, we discussed only two types of Markov models which are commonly used. The first one is discrete-time Markov chain (DTMC) and another is continuous-time Markov chain (CTMC).

(1)Discrete Time Markov Chains (DTMC) A countably infinite sequence, in which the chain moves state at discrete time steps, gives a DTMC. Let $T \in 0, 1, 2, ...$ is the parameter space and $P = [p_{ij}]$ is a transition probability matrix. It is a stochastic matrix since the sum of all elements in a row of P is 1 [50]. The state probability vector at time step n denoted by $\pi(n)$ [140] can be iteratively computed using:

$$\pi(n) = \pi(n-1)P \tag{2.1}$$

In terms of initial probabilities $\pi(0)$

$$\pi(n) = \pi(0)P \tag{2.2}$$

where, $P(n)$ is called n-step transition probability matrix of DTMC. Let $p_{ij}(n)$ is $(i,j)^{th}$ entry of matrix $P(n)$, represents the probability of reaching state j at time step n, starting from state i. Markov chains are of two types (1) Irreducible: if every state can be reached from every other state. (2) Absorbing: if there is at least one state i, there are no outgoing transitions. In the case of irreducible DTMC, the dominant metric is the probability of being in state i at time step n and in steady-state [49]. The steady state probability vector at time step n can be computed using Equation 2.3. To compute the steady state probability vector in the steady state, limits are taken on both the sides of Equation 2.2. This results in following system of equations for computing the probability vector of the system in the steady state

$$\pi = \pi.P \tag{2.3}$$

In case of an absorbing DTMC, there are three metrics of interest for state i:

1. the probability of being in state i at time step n
2. the probability of being in state i in the steady state.
3. expected number of visits t each one of the non-absorbing states i.

Let P be the transition probability matrix of an absorbing DTMC with total $'S'$ states where $'A'$ are absorbing states. Let the non-absorbing or transient states be labeled $1, ..., S-A$, and the absorbing states be labeled as $S-A+1, ..., S$. The transition probability matrix P of an absorbing DTMC can be partitioned as:

$$P = \begin{bmatrix} Q & C \\ O & I \end{bmatrix}$$

where, Q is an $(S-A) \times (S-A)$, I is an Identity matrix, O is an $A \times (S-A)$ matrix of zeros, and C is an $(S-A) \times A$ matrix. Let $F(n)$ denote the probabilities of being absorbed in state j starting from the transient state i in n steps. F is given as

$$F = \sum_{l=0}^{n} Q^l C \tag{2.4}$$

For steady state probability, $l \to \infty$

$$F = \sum_{l=0}^{\infty} Q^l C = (I-Q)^{-1} C \tag{2.5}$$

$(I-Q)^{-1}$ is called fundamental matrix M and is given by:

$$(I-Q)^{-1} C = I + Q + Q^2 + ... = \sum_{l=0}^{\infty} Q^l \tag{2.6}$$

(2) Continuous Time Markov Chains (CTMC) It is used in situations where the transitions between the states do not occur at specific time steps as in the DTMC [6]. In the time interval dt, the transition probabilities between states i and j are given as

$$P_{ij} = \lambda_{ij} dt \tag{2.7}$$

where,

λ_{ij} is the constant conditional failure intensity or failure rate which is defined as the probability of component failure per unit time. It is reciprocal to mean time to

failure, hence it is possible transition rate becomes zero. The transition matrix can be defined as [140].

$$P = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1m} \\ P_{21} & P_{22} & \cdots & P_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ P_{m1} & P_{m2} & \cdots & P_{mm} \end{bmatrix}$$

where,

P_{ij} is the transition probability between state i and j

$$= \begin{bmatrix} 1 - \sum_{k=2}^{m} \lambda_{11} dt & \lambda_{12} dt & \cdots & \lambda_{1m} dt \\ \lambda_{21} dt & 1 - \sum_{k=1,k\neq 2}^{m} \lambda_{2k} dt & \cdots & \lambda_{2m} dt \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{m1} dt & \lambda_{m2} dt & \cdots & 1 - \sum_{k=1,k\neq m}^{m} \lambda_{mm} dt \end{bmatrix}$$

With the transition matrix derived, the probabilities being in a specific state after a given time is given by $Q_j(t + dt)$, for state j at time $t + dt$.

Consider a two state Markov model as shown in FIGURE 2.2 with initial condition $Q_1(0) = 1$ & $Q_2(0) = 0$, it follows that

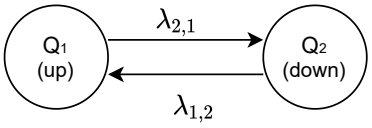

FIGURE 2.2: Simplex system model example

$$Q_2(t + dt) = \lambda_{12} dt Q_1(t) + (1 - \lambda_{21} dt) Q_2(t) \tag{2.8}$$

At any time, model has

$$Q_1(t) + Q_2(t) = 1 \tag{2.9}$$

On solving Equation (2.8) and (2.9),

$$Q_2(t+dt) = \lambda_{12}dt(1-Q_2(t)) + (1-\lambda_{21}dt)Q_2(t) \tag{2.10}$$

$$\frac{Q_2(t+dt)-Q_2(t)}{dt} = \lambda_{12}dt - (\lambda_{1,2}+\lambda_{21})Q_2(t) \tag{2.11}$$

$$Q_2(t) = \left(\frac{\lambda_{12}}{\lambda_{12}+\lambda_{21}}\right)(1-e^{-}(\lambda_{12}+\lambda_{21})dt) \tag{2.12}$$

On generalizing the above two state model into a model with n states, the probability of the system being in state i at time t yield

$$Q_i(t) = \left(\frac{\lambda_{incomingedges}}{\lambda_{incomingedges}+\lambda_{outgoingedges}}\right)(1-e^{-}(\lambda_{incomingedges}+\lambda_{outgoingedges})) \tag{2.13}$$

2.1.4.2 Petri Net

Petri net is a graphical as well as a mathematical tool that can be applied to model the distributed, non-deterministic, parallel, asynchronous systems. It was developed by Carl Adam Petri in 1962. It is a state-space based model. A Petri net (PN) is represented as a directed bipartite graph with two disjoint sets of nodes: Places P and Transitions T. Place node, modeled as a circle, represents the state or condition. The transition node, modeled as a bar, represents the discrete event or function. A transition is connected with the specific number of input and output places to express the pre and post conditions of the event. The system behavior is described in the form of possible system states (marking) and their transitions, which are graphically represented as tokens, a non-negative number of dots.

$PN = (P, T, A, M_0)$ is defined as a 4-tuple [103]

where,

$P = \{p_1, p_2, ..., p_n\}$ is set of places,

$T = \{t_1, t_2, ..., t_m\}$ is set of transitions,

$A \subseteq \{P \times T\} \cup \{T \times P\}$ is set of input and output arcs,

$M_0 = \{\mu_1^0, \mu_2^0, ..., \mu_n^0\}$ is initial marking,

$M : P \to N$ where $M(P_i) = \mu_i$ for $i = 1, 2...n$.

2.1.4.3 Stochastic Petri Net

Stochastic Petri Net (SPN) is an extended form of PN to model the stochastic processes. Here, each transition is associated with a positive, exponentially distributed random variable that represents delay from enabling to firing that particular transition. Formally, an $SPN = (P, T, A, M_0, \lambda)$ is defined as a 5-tuple [98] where

$P = \{p_1, p_2, ..., p_n\}$ is set of places

$T = \{t_1, t_2, ..., t_m\}$ is set of transitions

$A \subseteq \{P \times T\} \cup \{T \times P\}$ is set of input and output arcs

$M_0 = \{\mu_1^0, \mu_2^0, ..., \mu_n^0\}$ is initial marking

$M : P \to N$ where $M(P_i) = \mu_i$ for $i = 1, 2...n$

$\lambda = \{\lambda_1, \lambda_2, ...\lambda_m\}$ is set of firing rates which is reciprocal to the average firing delay d_m associated with transitions t_m .

If several transitions are enabled, a transition with the shortest delay will get priority in the firing. The elementary SPN models are shown in FIGURE 2.3. Due to the memoryless property of the exponential distribution of firing delay, the reachability graph of SPN can be converted into a finite Markov Chain as shown in FIGURE 2.4, where λ_0 is the firing rate associated with transition T_0. Thus, SPN combines the power of PN and Markov processes. This may be useful to calculate different

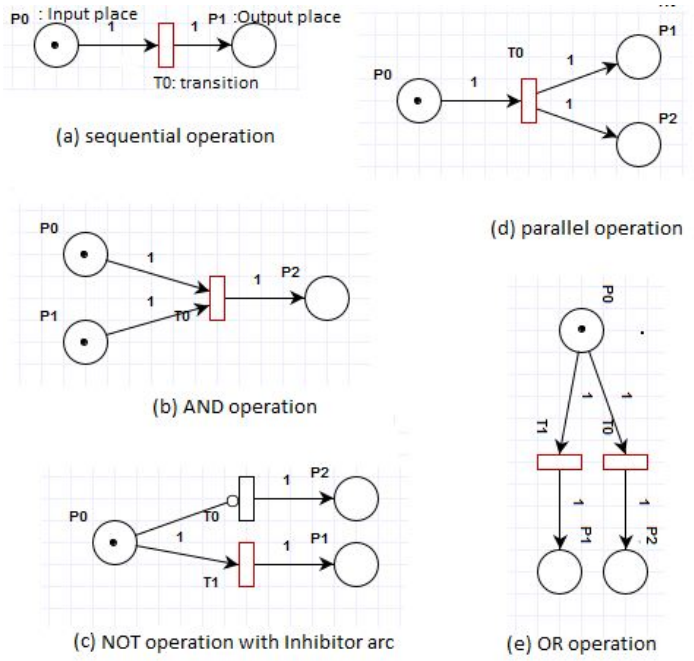

FIGURE 2.3: Elementary SPN models

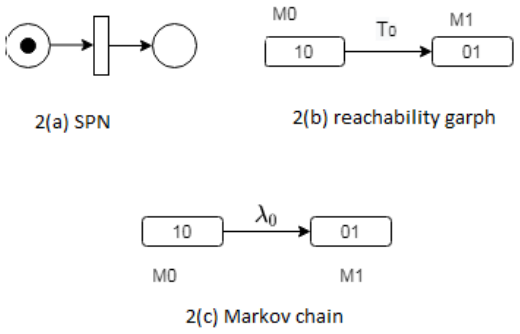

FIGURE 2.4: Transformation of SPN model

performance measures of system to analyze the dynamic behavior of CPS under normal condition, attack and applied mitigation.

2.1.4.4 GSPN

GSPN, an extension of Petri Net SPN, is well suited to model the non-deterministic distributed, asynchronous systems with uncertainty. The transitions are divided into immediate and timed transition sets, denoted by a solid and empty bar as shown

in FIGURE 2.5 where T1, T2 are immediate transitions and T0, T4, T5 are timed transitions.

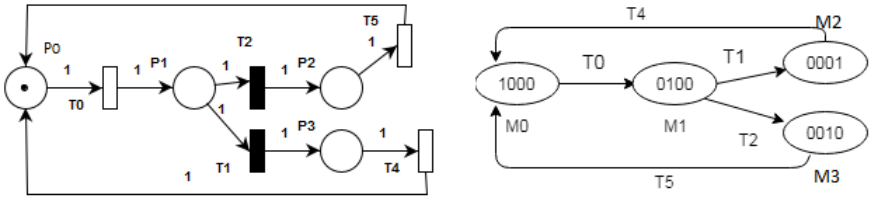

FIGURE 2.5: GSPN model and its reachability graph

The dynamic behavior of the Petri net is shown as marking that associate tokens, a non-negative number of dots, with each place. Formally, A $GSPN = (P, T, A, M_0, \lambda)$ is defined as a 5-tuple [27] where

$P = \{P_0, P_1, ..., P_{n-1}\}$ is set of places

$T = \{T_0, T_1, ..., T_{m-1}\}$ is set of transitions

$T = T_{td} \cup T_{id}$ and $T_{td} \cap T_{id} = \phi$, where

$T_{td} \subseteq T$ is a set of timed transitions and,

$T_{id} \subset T$ is a set of immediate transitions,

$P \cap T = \phi$ and $P \cup T \neq \phi$

$A \subseteq \{P \times T\} \cup \{T \times P\}$ is set of input and output arcs

$M_0 = \{\mu_0^0, \mu_1^0, ..., \mu_{n-1}^0\}$ is initial marking

$M : P \to N$ where $M(P_i) = \mu_i$ for $i = 0, 1...n-1$

and N is a set of natural numbers

$\lambda = \{\lambda_0, \lambda_1, ...\lambda_{m-1}\}$ is set of non negative real numbers

where,

$$\lambda_k = \begin{cases} \text{firing rate} & \text{if } T_k \in T_{td}, \\ \text{firing weight} & \text{if } T_k \in T_{id} \end{cases}$$

Thus, exponentially distributed firing delay is associated with each timed transition as compared to immediate transition that has negligible firing delay. Although, transition weights are associated with immediate transitions for setting the firing priorities among them. To study the stochastic behavior of system, a reachability graph is generated from the GSPN model as shown in FIGURE 2.5, which is transformed into an Embedded Markov Chain (EMC). If for a marking M_j, multiple timed transitions are enabled which is denoted as $E(M_j)$, the firing probabilities of one of the enable transition T_k is given as

$$p\{T_k|M_j\} = \frac{\lambda_k(M_j)}{\sum_{i:T_i \in E(M_j)} \lambda_i(M_j)} \tag{2.14}$$

To handle the state explosion problem, EMC can be transformed into Reduced Embedded Markov Chain (REMC). The REMC is constructed with tangible states ($\tilde{T}$) of reachability graph after removing vanishing states ($\tilde{V}$) (as the time consumed at $\tilde{V}$ is 0 due to negligible firing delay of immediate transition) to calculate the probability of occurrence an event.

2.2 Literature Review

In this section, we perform the literature survey of existing modeling, analysis and system organization methods and try to find the associated issues with these existing methods. While doing that, to ease out the survey process, we have divided the survey into two parts. The first part includes the existing modeling and analysis methods for CPS security, and second part presents the organization and management methods of CPS.

2.2.1 Modeling and Analysis Methods for CPS Security

The section presents a literature review of significant threat modeling and mitigation studies. There is plenty of literature on security concerns based on different

mechanisms, including data-driven and model-based approaches. Even as a part of model-based approaches, many combinatorial and state-space based methods are proposed for system security modeling and analysis.

Nandi *et al.* [105] presented attack graphs to identify cyber security threats. These graphs attempt to enumerate vulnerabilities and penetration paths and further transformed the problem as linear programming to provide an algorithmic solution based on a heuristics approach to find an optimal solution. However, the authors did not prove the correctness of the algorithm. Moreover, this model is not considering the randomness and stochastic nature of security events and systems.

The authors [1] combined bowtie analysis with attack tree for safety-security risk analysis of industrial control the system, Although the model does not present any quantitative solution for threat assessment and security mitigation.

The authors [32] performed a cost-benefit analysis based on attack trees from the system administrator's perspective. The attack defense tradeoff is formulated as arm-race multi-objective optimization for optimal security hardening.

The authors [121] embedded a fuzzy set theory with an attack tree to deal with uncertainty. The model analyzes security risks based on attack path probabilities. Although the approach is qualitative and the model cannot identify several behavioral constructs of a system disturbed by attackers, like liveness, safety, boundedness, consistency, and recoverability.

However, the attack model is limited to assessing just one attack (top event) and not suitable for multi-state variable modeling and assessing various output variables in the same model. This limitation is overcome by Bayesian networks that comparatively increases its importance in literature.

Marrone *et al.* [93] extended UML to model attack and protection concerns in critical infrastructures. The model is transformed into the Bayesian network for vulnerability analysis based on estimated probability distribution. BN supports multi-state variable modeling and several output variables.

Munoz *et al.* in [102] considered the attacker's uncertain behavior and proposed an inference algorithm using Bayesian networks for the attack graph to perform static and dynamic security risk analysis. The goodness of the algorithm is also shown in terms of memory and speed.

The authors [145] employed BN with the FAIR model for cyber-security risk analysis and decision making. The model performance is evaluated using Monte Carlo simulation. However, no methodology is presented in the literature that performs security analysis qualitatively and quantitatively to verify the structural and behavioral properties.

The combinatorial models have limitations in the expression of system stochastic behavior. While these methods are quite popular among system security analysts, the classical formulation cannot capture the dependencies of security vulnerabilities on the sequencing of events and system behavior.

State-space models are more comprehensive and well known for critical system verification and analysis. They allow the modeling of complex relationships where the transition structure encodes the sequencing information and dependencies. These methods have been used as mathematical models that specify probabilistic assumptions about transition behavior and time durations. Some of the existing work based on these methods are as follows-

Zurawski *et al.* [158] introduced the industrial applications of Petri nets. The work studies the behavioral properties to analyze model performance, including reachability, boundedness, safety, conservation, and liveness. However, it does not include other properties like consistency, synchronic distance, and fairness, which can be useful for more in-depth model evaluations.

Chen *et al.* [25] modeled the cyber and physical attacks on the smart grid using Hierarchical Petri nets. The approach overcomes the modeling complexity of large CPSs. Although, the authors just focused on modeling constructs while behavioral analysis is entirely missing.

Cho *et al.* [28] presented security and dependability modeling of CPS using generalized stochastic Petri net, where cyber and physical security threats are considered to penetrate the control system. The security analysis is performed by just considering reachability criteria. The remaining metrics are not considered that are useful for verifying model correctness.

Marashi *et al.* [92] analyzed the security using Aspect-oriented Petri net where vulnerabilities are identified, and countermeasures are applied as aspects. However, the model lacks the behavioral metrics that are useful for in-depth model analysis.

Fu *et al.* [39] evaluated the effect of firewall and password model on intrusion attacks using Petri net. The model finds out steady-state probabilities and mean cycle time of attack defense net to analyze the effectiveness. The effect of an attack on system behavioral properties like deadlock situation, ability to recover, resource exhaustion, resource conservation, and model correctness is missing.

Along with this, several works have been done that focus on the selection of threat mitigation methods and analyzing their impact on system security to reduce cyber risks.

Madan *et al.* modeled the intrusion tolerance system's response using graph-based stochastic modeling techniques in [90]. The work quantified the security attributes (*i.e.* mean time to security failure and security failure probability) to analyze the effect of intrusion tolerance. Although, the proposed model does not explicitly analyze the impact of attack defense on CPS. How the system can be attacked and what kind of responses need to be applied is not explained thoroughly. The transition probabilities are assumed randomly without considering the strength of defenses. Moreover, the graph-based stochastic model may suffer from a state explosion problem.

Xu *et al.* [149] proposed Aspect-oriented Petri nets for modeling and verifying system functionality, threats, and mitigation to design secure software. The proposed model is deterministic and does not consider the possibility of mitigation failure. Thus, this is not well suited for real-time SC-CPSs.

Ten *et al.* [133] presented a GSPN based model that assesses the cyber-security risk by identifying the vulnerabilities in Supervisory Control and Data Acquisition System (SCADA) and integrates firewall and password systems to handle the security vulnerabilities.

Ramos *et al.* [114] presented an SPN model to evaluate physical security using multiple hardware protection to delay the power system's failure. Although, the validation is not mentioned.

Cavusoglu *et al.* [22] used game theory to analyze the combined effect of firewall and IDS. Although, the work only focused on network-level vulnerabilities.

Tjao *et al.* [134] presented risk-aware business process modeling where the influence of threats and different safeguards on resources and activities is demonstrated.

Cho *et al.* [28] extended the work of [133] to present security (physical, cyber) and dependability evaluation model for NPP using GSPN. In both the work, the primary focus is on preventive measures based on the firewall and password model.

Mitchell *et al.* [96] discuss several types of failures, including exfiltration, attrition, and perversion failure. The authors proposed an SPN-based model to analyze the effect of intrusion detection and response on these failures and revealed that adjusting the strategy and strength of IDS enhance the cyber-physical system reliability.

Yang *et al.* [151] presented an intrusion detection system for IEC 61850 based SCADA, which is a responsive measure to mitigate the cyber threat.

shin *et al.* [125] used Event tree and Bayesian networks to evaluate the cyber risk for the nuclear industrial control system. They do not present any formal definition of attack, defenses, and combined effects.

Orojloo *et al.* [109] presented an approach to evaluate the direct and indirect impact of cyber attacks on physical processes by calculating the deviation from the normal condition. This approach helps to rank the critical assets of a CPS and prioritize

the attacks for retaliation. As a defensive countermeasure, the authors just mentioned the use of IDS, network segmentation, VPN network, and defense-in-depth strategies. Although, no quantitative analysis is performed to verify the effect of these mitigation strategies.

Fu textitet al. [39] evaluated the effect of firewall and password model on intrusion attacks using Petri net. Although, the model missed the need for authorization to

TABLE 2.1: Related work

Proposed work	**Objective**	**Modeling approach**	**Merit**	**Limitation**
Abdo *et al.* [1]	risk analysis	combined bowtie analysis with attack tree	safety-security risk analysis of industrial control the system	model does not present any quantitative solution for threat assessment and security mitigation.
Kleinmann and Wool [70]	Anomaly detection	State chart DFA	used to visualize, specify, build; Easy to understand	unable to model stochastic behviour
Nandi *et al.* [105]	to identify cybersecurity threats using 'what-if' constructs	Attack graph	these graphs enumerate vulnerabilities and penetration paths based on heuristics	unable to quantify randomness and stochastic nature of security events
Shang *et al.* [121]	to analyze security risks based on attack path probabilities	Attack tree and fuzzy sets	embedded a fuzzy set theory with an attack tree to deal with uncertainty	attack model is limited to assessing just one attack (top event) and not suitable for multi-state variable modeling and assessing various output variables in the same model
Huang *et al.* [58]	proposes a dynamic game framework to model a long-term interaction between a stealthy attacker and a proactive defender	Game theory	The stealthy and deceptive behaviors are captured by the multi-stage game with incomplete information via observations and learning.	can not be used to model the impact of attacks on process flow
Marrone *et al.* [93]	vulnerability analysis	Bayesian network	model attack and protection concerns in critical infrastructures	unable to handle parameter uncertainty and dynamic behavior of system
Munoz-Gonz- alez *et al.* [102]	security risk analysis	Bayesian networks	considered the attacker's uncertain behavior and designed an inference algorithm to find attack paths	unable to handle parameter uncertainty and dynamic behavior of system
Marashi *et al.* [92]	security analysis	Aspect-oriented Petri net	vulnerabilities are identified, and countermeasures are applied as aspects to improve modularity	However, the model lacks the behavioral metrics that are useful for in-depth model analysis.
Chen *et al.* [25]	vulnerability analysis	Hierarchical Petri nets	the approach overcomes the modeling complexity of large CPSs	focused only on modeling constructs while behavioral analysis is entirely missing
Fu *et al.* [39]	security analysis	Petri net	quantitative evaluation of firewall and password model effect on intrusion attack probabilities	no provisions of handling these preventive measures failure, state space explosion
Marashi *et al.* [92]	security analysis	Aspect-oriented PN	vulnerabilities identification and countermeasures application as aspect for modular design and maintainability	model lacks the behavioral metrics useful for defense in-depth analysis, state space explosion
Ten *et al.* [133]	vulnerability assessment	GSPN	Presented firewall and password model	no security provisions for access control vulnerabilities and responsive actions
Cho *et al.* [28]	security analysis	GSPN	Proposed firewall and password as cyber security model with physical security	no security provisions for access control vulnerabilities and responsive actions
Cho *et al.* [151]	intrusion detection system	knowledge based system	an IDS based responsive measure for IEC 61850 based SCADA	no security provisions to prevent intrusion attempts
Mitchel *et al.* [96]	intrusion detection model	GSPN	detect and respond to the attack that has already damaged the system nodes	no provisions are discussed to prevent the intrusion attempts.
Shin *et al.* [125]	risk assessment	Event tree and Bayesian networks	evaluates the cyber risk for the nuclear industrial control system	do not present any formal definition of attack, defenses, and combined effects.

decide the privileges. Moreover, this model only talks about preventive measures and does not discuss the role of reactive measures for better security implementation.

The authors [26] presented cyber-physical execution semantics to defend against data-oriented attacks. Event-aware finite automata is used for behavior modeling of CPS that detects the existing anomalies. This model also focuses only on attack detection.

The authors [138] presented an anomaly-based intrusion detection model to detect and respond to the attack on NPP. However, no provisions are applied to harden the protection and resist intrusion attacks.

The authors [58] presented a proactive security mechanism based on the dynamic game approach to increase attack cost and reduce the cyber risk of CPS. The deceptive and stealthy behavior is captured by a multi-stage game where both attacker and defender policies are predicted by computing the Bayesian Nash equilibrium.

By reviewing the existing works, two research gaps are identified.

(1) There is a need for interaction modeling between the digital control system and physical processes (i.e., the value of physical process variables) and behavioral analysis of the system under threat.

(2) No work has been done yet that provides an approach of analyzing the effect of integrating preventive and responsive defense measures to secure SC-CPS. Hence, we present an evaluation approach for security measures to select alternative designs for better system protection.

2.2.1.1 Organization and Management of Distributed Secure CPS

As the concept of smart cities is being developed as a CPS, Jalali *et al.* presented a three-layer architecture [61] for a smart city. The architecture includes the sensory, network, and control & service layers, discussing supporting technology for each layer.

To manage the generated data in smart cities, in [44], Gaur *et al.* proposed a semantic web technology-based multi-level architecture for a smart city.

The architecture consists of data to service transformation layers such as data collection, data processing, data integration and reasoning, device control & alerts. In [78], the authors presented a more detailed and classic $5C$ CPS architecture which consists of connection, conversion, cyber, cognition, and configuration layers to optimize CPS roles and functions for manufacturing industries.

Next, JR Jiang extended the $5C$ architecture proposed in [78] and presented it as $8C$ architecture [63] by adding customer, coalition, and content for broader adoption in industries. However, the authors did not mention the management procedure of these architectures [44] [61] [63] [78].

To reduce latency, monitor network traffic and reduce system management complexity, in [85] and [8], Liu *et al.* and Balta *et al.* presented two centralized architectures of CPS. However, centralized architecture increases the risk of a single point of failure.

To deal with these challenges, Garofalo *et al.* presented a concept of a decentralized real-time system [43]. They applied the decentralized system to control urban drainage networks equipped with multiple sensors and a series of actuators. Moreover, the authors presented a gossip-based algorithm for achieving performance and fault-tolerance properties. However, there is a lack of provisions in [78] [44] [61] [85] [8] [63] [43] to make the system secure.

In [120], the authors proposed a hybrid smart city cyber security architecture to analyze the threats and associated risks. To deal with security concerns in widely adopted networked and web-accessible CPSs, Zhu *et al.* presented a hierarchical architecture [157] for dealing with cross-layer CPS security. They applied game theory to evaluate the effect of possible strategies of attackers and defenders on system security. However, this is not a unified architectural model integrating functionality and security. There is no provision of being fault-tolerant.

Tao *et al.* presented a cloud-based multi-tier architectural model [132] to enable interactions among different heterogeneous devices for IoT-based smart homes. Moreover, ontological constructs integrate security and privacy in the interaction process. Although the cloud supports the distributed architecture, the presented architectural model is managed in a decentralized manner but not in a purely distributed manner at the cyber level, limiting the model's breach tolerance and fault-tolerance capabilities.

To facilitate secure data communication, Vandana *et al.* presented SDN-based centralized architecture [142] for IoT to ensure secure data communication. SDN can detect anomalies and ensure some primary inhibition of communication network attacks. The SDN based paradigm, in essence, describes a centralized control architecture where applications (the S in SDN) possess the intelligence of the system and fulfill many roles such as computing, decision making, and reconfiguration (of devices) while leveraging the global view provided by a (logically) centralized controller. However, centralized architecture suffers from a single point of failure.

In [88], Liu *et al.* presented SDN-based data transfer security model 'middlebox-guard' to manage the data flow through SDN with defined security policies. They mainly focused on the selection of the appropriate location of middlebox deployment and presented the algorithmic solution for the same. Although, it is neither a unified architectural model to organize functionality with security nor a fault-tolerant model.

In [59], the authors improved [142] by presenting distributed architecture as Black SDN-IoT for smart city. The architecture integrates the NFV to apply device virtualization and monitor traffic flow. However, the main focus in SDN-based approaches [142, 59, 88] is on the network layer and traffic security only, where security is the sole responsibility of the SDN controller. In this scenario, if the security controller of the SDN controller fails, the system security gets compromised. There is no mechanism for selecting the appropriate security controller node immediately.

Lawal *et al.* presented real-time detection, and mitigation approach of distributed denial of service attack on SDN [75]. However, the approach is not fit for large CPS. Moreover, the work does not provide any architectural or design solution for separating the functional and security concerns for CPS.

In [154], the authors proposed a distributed intrusion detection system applied in multiple layers, including home area network, neighborhood area network, and wide area network for smart grid.

Feng *et al.* considered connected and automated vehicles (CAVs) as distributed CPS and proposed a design for intelligent transport systems using information graphs [36]. The proposed design point out the security requirements and use edge computing to process the information locally. However, the authors do not provide a methodology to integrate and analyze the security measures with functionality.

In [77], Lee *et al.* suggested a distributed architecture to overcome the centralized industrial network, security, and trust issue of CPS. They suggested that the security distribution should be at sensor level and computing level to take advantage of distributed computing in handling the performance and privacy concerns. However, the authors do not present any explicit explanation or in-depth methodology to organize and coordinate the functionality and security.

In [147] [146] and [148] the authors proposed the methodologies for privacy protection and handling the trust issues in information retrieval services hosted on cloud. These works present different algorithms to construct ideal dummy queries to meet the privacy model. However, these approaches are not designed for cyber-physical systems' privacy and security.

Next, Liu *et al.* proposed hierarchically distributed intrusion detection for anomaly detection in industrial CPS [86]. The framework applies anomaly monitoring methods at each layer of CPS, including perceptual layer, data transmission layer, and application control layer.

TABLE 2.2: Comparative analysis with existing works

Existing Work	System Management	Target Work	Limitations	Security arrangement	Fault-tolerance	Performance	Maintainability
Gaur *et al.* [44]	not discussed	presents semantic web technology driven CPS architecture	not a unified architecture for integrating functionality and security	not considered	not discussed	not discussed	not considered
Lee *et al.* [78]	not discussed	CPS reference architecture to describe its layers and roles	not a unified architecture for integrating functionality and security	not considered	not discussed	not discussed	not considered
Jalali *et al.* [61]	not discussed	three-layer architecture for a smart city	not a unified architecture for integrating functionality and security	partial	not considered	not evaluated	not considered
Zhu *et al.* [157]	not discussed	hierarchical architecture with game theory to deal with cross-layer CPS security	not a unified architecture for integrating functionality and security	considered	not discussed	not evaluated	not considered
Liu *et al.* [85]	centralized manner	software-defined IoT architecture for smart urban sensing where Centralized controllers are designed to manage physical devices and provide APIs of data acquisition, transmission, and processing services	not a unified architecture for integrating functionality and security. Moreover, the risk of a single point of failure	not considered	low due to the risk of single-point failure	not evaluated	not considered
Vandana [142]	centralized network control	SDN-based architecture for IoT to ensure secure data communication using the inherent capability of SDN controllers	only focus on the network layer and traffic security where security is the sole responsibility of the SDN controller. Moreover, the risk of a single point of failure	considered	low due to the risk of single-point failure	low due to bottleneck	not considered
Balta *et al.* [8]	centralized manner	centralized framework for system-level control and management of additive manufacturing fleets	not a unified architecture for integrating functionality and security, Moreover, risk of a single point of failure	not considered	ow due to risk of single-point failure	not evaluated	not considered
Garofalo *et al.* [43]	decentralized manner	gossip-based algorithm for achieving performance and fault-tolerance properties	not a unified architecture for integrating functionality and security	not considered	considered	considered	not considered
Tao *et al.* [132]	decentralized manner	a cloud-based multi-tier service-oriented architectural model with ontological constructs for interactions among different heterogeneous devices for IoT-based smart home	who is coordinating the task distribution and aggregation is not discussed	considered	no explicit arrangement is presented	not evaluated	considered
Sengan *et al.* [120]	decentralized manner	hybrid smart city cyber security architecture to analyze the threats and associated risk	not a unified architecture for integrating functionality and security concerns	not considered	not considered	not evaluated	not considered
Islam *et al.* [59]	distributed manner	distributed architecture as Black SDN-IoT with SDN controller, device virtualization to control and monitor the traffic data flow	only focus on the network layer and traffic security. Moreover, not a unified architecture for integrating functionality and security concerns	considered	high	analysed	medium
Lee *et al.* [77]	distributed manner	emphasized the need for distributed architectural management, where security distribution should be at sensor level as well as computing level to take advantage of distributed computing in handling the performance and privacy concerns	not a unified architecture for integrating functionality and security concerns. Moreover, no discussion on how to coordinate, distribute and aggregate the functional and security tasks	considered	no explicit arrangement is presented to achieve it	not evaluated	not considered

Similarly, Satam *et al.* [119], the authors present a security framework to defend against cyberattacks for IoT, where the intrusion detection system is applied for IoT sensors network and Bluetooth protocol. The IDS detects cyber-attacks based on extracted features of Blue-tooth and sensor signals, which are further used by different machine learning classifiers. However, these works [154, 86, 119] do not consider the scenario where the security nodes may also be failed or be compromised by sophisticated and coordinated attacks.

Table 2.2 presents a comparative analysis of existing architectural arrangements works and philosophy.

Therefore, to the best of our knowledge, no work has been done yet that presents a distributed architectural model to integrate and organize security with functionality in existing CPSs. Moreover, who will coordinate the activity among heterogeneous nodes in CPS? If the coordinator node is to be chosen, then how to elect that as the existing leader election algorithms [110, 11, 113, 14, 12] are not suitable for a large CPS as the proposed work elect a general leader without considering the need of functionality and security requirements of time-constrained real-time systems. How to implement adaptive functionality and security arrangements in case functional or security nodes are compromised by sophisticated and coordinated attacks for CPS? These are still open challenges that are not dealt with by the community. Hence, in Chapter 5, we present a distributed architectural model to coordinate and integrate the functionality and security, avoid a single point of failure, and increase fault tolerance at reduced communication latency in a CPS by bringing in the concept of fault-tolerant security and functionality leaders, unlike Chapter 3 which provides a systematic approach to model and analyze the stochastic behavior of CPS in the presence of threats and possible mitigations in the early stages of system development, and Chapter 4 which presents a GSPN based approach is presented to analyze the effect of combining the preventive and reactive measures against cyber-attacks and handling the state space explosion problem. By analyzing the model, system security is quantitatively estimated in terms of mean-time-to disrupt and system availability.