Securing IoT-based Cyber-Physical Human Systems against Collaborative Attacks

Sathish A.P Kumar[1], , Bharat Bhargava[2] , Raimundo Macêdo[3] and Ganapathy Mani[2]
[1]Coastal Carolina University, Conway, SC, USA
[2]Purdue University, West Lafayette, IN, USA
[3]Federal University of Bahia, Ondina, Salvador, Bahia, Brazil

***Abstract*—Security issues in the IoT-based Cyber-Physical Systems (CPS) are exacerbated with human participation in CPHS due to the vulnerabilities in both the technologies and the human involvement. A holistic framework to mitigate security threats in the IoT-based Cyber-Physical Human Systems (CPHS) environment is presented to mitigate these issues. We have developed threat model involving human elements in the CPHS environment. Research questions, directions, and ideas with respect to securing IoT-based CPHS against collaborative attacks are presented.**

Keywords— Security; Cyber-Physical Human Systems; Internet of Things; Threat Index, Byzantine replication; Intrusion Prevention; Intrusion Detection; Intrusion Tolerance

I. INTRODUCTION AND BACKGROUND

Integration of cyber and physical components has led to the development of new critical complex systems called Cyber-Physical Systems [14]. Internet of Things (IoT) involves the deployment of these CPS [17]. Human participation extends it to a Cyber-Physical-Human Systems [1, 4]. A human can act as the controller of CPHS. As human behavior is difficult to model, understanding, validating, and protecting such systems is challenging. CPHS are vulnerable to threats such as spam and phishing attempts, domain name system exploitations, replication attacks, and denial of messages attacks, where malicious cyber components and error prone and malicious humans prevent some honest nodes from receiving broadcast messages [3, 4]. Many safety-critical applications of IoT-enabled CPHS must cope with a large number of heterogeneous devices and systems with distinct computing and communication capabilities [23]. Such physical, human, and cyber interactions can result in system behaviors, which cannot be precisely anticipated at system design. Guaranteeing dependable, secure, and timely system operations during runtime is a design challenge. Characterizing, detecting, and mitigating the dynamic nature of the human involvement in the context of collaborative attacks in the IoT-enabled CPHS needs further research. Collaborative attacks are a class of attacks where multiple malicious adversaries including the humans with malicious intent collude. Human actions interleave and synchronize to accomplish disruption, deception, usurpation, or data disclosure against IoT-enabled CPHS entities. Collaboration among human entities with malicious intent can mount sophisticated CPS attacks. When attacks result in the complete control of a system component, detecting, and recovering from these security faults is hard since the compromised component can behave in byzantine manner.

Since the devices in the CPH environment are connected to the Internet and critical data is associated with them, there are concerns about the security [24]. By security, we mean the degree of resistance to or protection of the IoT infrastructure based CPH applications. Many of these devices are easy targets for intrusion because they rely on very few outside resources and are often left unattended [26]. Once the network layer is compromised, it is easy for a hacker to gain control and maliciously use a device and attack other adjacent devices in neighborhood through the original compromised node [32]. In particular, appliances that maintain an online presence are easy to attack. These devices that do not have any virus protection or malware protection are highly susceptible to being used as "bots" to forward malicious code to infect other devices [27, 29, 30, 31]. The International Data Corporation predicts that more than 50 billion devices will be connected in the IoT framework by the year 2020, with a good amount of these being appliances. There is ample opportunity for hackers to use these devices to their advantage through "denial of service" attacks, malicious email, and other harmful worms or Trojans [33].

Protecting privacy of the human entities in CPHS environment is another related issue that needs to be addressed. We intend to evaluate how we may be able to model attacks on the privacy using above causal relationships among events. To illustrate it, let us consider the example of Alice who provides her phone number on a staff directory of a CPHS environment. Another user Bob may contact her on her phone number and may obtain her role and office address. At this point Bob may use any word processing software to store this information. Now, without Alice's knowledge an unknown third party holds two key identity attributes about Alice. Furthermore, if some of Alice's friends decide to wish her on her birthday using a medium which is publicly visible (e. g. Facebook wall, which is visible to all friends), then even if Alice may not have publicly disclosed her birthday, others will be able to infer this. Once we generate the causal event graph it is clear where which events and paths may lead to situations where privacy may be compromised. At each point of the graph it may be possible to identify what is the possible loss of privacy and the results may be used to alert the user to avoid taking such paths, or to take corrective measures. This approach will be applied to online social networks where users who may be interactive human actors of CPHS environment tend to provide a plethora of a subset of personal information coupled with other multimedia artifacts such as videos and photographs.

Rest of the paper is organized as follows: Section 2 presents the motivation and rationale for our research approach. Section 3 presents our proposed security framework to secure CPHS environment against collaborative attacks. Section 4 presents our threat modeling approach involving human entities. Finally, section 5 presents the conclusion.

II. Motivation and Rationale for the Approach

This proposed framework will mitigate security challenges that could result in the imminent loss of life or property in Cyber-Physical Human Systems. For example, in an intensive care unit (ICU), processing is time critical, human-centric and includes complex devices and software. In particular, seizure of the health information systems and services implemented in the low bandwidth IoT networks in the ICU environment mean risks of life-threatening situations and loss of business [24, 25]. The CPH workflow and IoT protocols are not only a function of the tasks and environment at hand, but must also be aware of the capabilities and training of the personnel involved [20]. The proposed security protocols will keep the IoT-based CPH environment secure from the actions of the personnel involved.

A CPH system operates to accomplish a mission under rapidly changing circumstances. The stressful, rushed, and often unfriendly environment of a CPH system means that possibilities for attacks are high [11]. This research offers security and resilience in the face of attacks exposed by the vulnerabilities. Effective and immediate intervention enabled by an optimized CPH system will dramatically reduce the risk due to security attacks.

Researchers have advocated the use of intrusion tolerance mechanisms that allow a system to keep working properly even when intruders control some of its components [2, 3, 15]. The malicious attacks are modelled as arbitrary or byzantine faults [5], and many byzantine fault tolerance mechanisms have been proposed [6, 7]. Research is needed to study explicitly the human interactions in various roles of a CPHS, which will require proper modelling and tools. Development of models of human role in addition to system modelling for effective automatic and self-adaptive security mechanisms is needed. This will allow for autonomously recovering the system functionality in a timely fashion, or degrading the system to a safe operation [8, 12, 13, 16]. We have applied such autonomic or self-adaptive mechanisms to implement several fault tolerant distributed protocols, such as failure detection and group communication [9, 10]. Though intrusion tolerance allows the system under attack to work properly, prevention of intrusions is needed to avoid further failures when system resources become depleted. Intrusion tolerance should be complemented with intrusion prevention and intrusion detection and they should work in a coordinated and integrated fashion [3, 15]. Providing such functionality to a system involves adding to it a number of structural and behavioral mechanisms, including component redundancy, path redundancy, data redundancy, decentralization and threshold cryptography, self-organization, dynamic routing, backward recovery and forward recovery, among others. A system [57] was built that provides support for antifrugality (increase in capability and robustness as a result of mistakes, faults, attacks, or failures), resiliency, and continuous availability under highly dynamic contexts involving cyber-attacks and service failures for applications. The system allows for dynamically reconfigured service composition based on changes in the context with respect to timeliness and accuracy of information as well as the type, duration, extent of attacks. It considers trust levels of services and the complexity of the environment and any operational context changes such as different platform, emergency, endpoint change etc. Our ultimate goal is antifrugality in IOT.

The framework and methods that are proposed will lead to practical guidelines for developing adaptive defense mechanisms that can deal with type, severity, timing, extent, duration, and collaboration present in malicious attacks in IoT enabled CPHS. In our earlier work with respect to defending against collaborative attacks on route discovery protocols, where multiple sites colluded by dropping the packets and pretended as if the data packets have been forwarded in MANET, we designed hash function methods that contain information from both data traffic and forwarding paths to detect the collaborative attacks [21, 58, 60]. Similarly, we proposed vulnerability analysis solutions to model and analyze attack graphs for the collaborative attacks in networks and Worldwide Interoperability for Microwave Access (WiMAX) protocol suite [55]. Experimental studies integrate ideas from networking, distributed systems, and security / privacy / reliability research will answer questions about performance and implementation of large CPHS under multiple attacks. This paper will contribute to the community by providing researchers with empirical parameters and observations that can be used in peer-to-peer, cooperative systems, and large-scale applications. In addition, the proposed concepts will lead to the science of understanding and dealing with collaborative attacks and coordinated defense through proper modelling of a faulty human component, byzantine fault tolerance, identity management (IDM), and autonomic, self-adaptive techniques to prevent, detect, and counter those CPHS attacks.

III. Proposed Security Framework

To address the issues and questions described in the previous section, we propose the framework as shown in Figure 1.

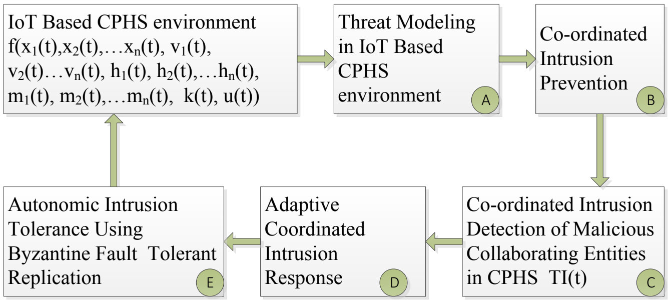
IoT Based CPHS environment
$f(x_1(t),x_2(t),\ldots x_n(t), v_1(t), v_2(t)\ldots v_n(t), h_1(t), h_2(t),\ldots h_n(t), m_1(t), m_2(t),\ldots m_n(t), k(t), u(t))$
Threat Modeling in IoT Based CPHS environment
A
Co-ordinated Intrusion Prevention
B
Co-ordinated Intrusion Detection of Malicious Collaborating Entities in CPHS TI(t)
C
Adaptive Coordinated Intrusion Response
D
Autonomic Intrusion Tolerance Using Byzantine Fault Tolerant Replication
E

Figure 1 – Security Framework for IoT-based CPHS Environment

Each step in the proposed framework is explained in the following sub sections.

A. *Threat Modeling in IoT-based CPHS Environment*

In this step, an understanding of the methods of operations of coordinated CPHS attacks on network and distributed database environments are explored from a holistic system perspective to understand, analyze, classify, and defend against collaborative CPHS attacks [59]. Since the addition of humans in the CPS framework creates a lot of uncertainty due to unpredictable human behavior, modeling the characteristics and methods of attacks due to humans is important. Research plan is to model malicious behavior of dynamic CPHS components, including humans

Threat models for IoT-based CPH will be constructed in this step. A module for looking up and collecting security attack data for the considered threat scenarios/models in IoT-based CPH system can be implemented in this step. IoT-based CPH is represented as a function: $f(x_1(t), x_2(t),\ldots x_n(t), v_1(t), v_2(t)\ldots v_n(t), h_1(t), h_2(t),\ldots h_n(t), m_1(t), m_2(t),\ldots m_n(t), k(t), u(t))$, where $x_n(t)$ represents the significant attack sensitive parameters, $v_n(t)$ represents the parameters which are not significant in representing the node vulnerability, $m_n(t)$ represents the mobility parameters, h(t) represents the human behavior parameters, k(t) represents the attack and u(t) represents the control input x_n'(t) represents the modified values of the significant attack sensitive parameter due to influence of the attack k(t) and the control input u(t).

We propose a methodology for identifying multiple human entities cooperating as a group by introducing two key mechanisms: 1) Data Routing Information (DRI) Table with device identities and their network connection information, which can help CPHS to keep track of any unusual behavior patterns of and 2) Cross Checking. The process of cross checking the intermediate entities is a onetime procedure which we believe is affordable to secure a network from multiple malicious human entities. The proposed solution has two new key advantages: 1) Identification of multiple collaborative human entities and 2) Discovery of secure source-destination paths that avoid human entities acting in cooperation. Using Fuzzy logic, causal model (discussed in the next subsection), and other machine learning techniques, we will address the following questions: (1) how to efficiently integrate information from multiple entities through IDM; (2) how to develop attack detection mechanisms that are robust against noises in the detection capabilities of the mechanisms; (4) how to determine the tradeoff between the detection frequency and information; (3) what is the relationship between the range and the dynamics in CPHS. Using machine learning techniques to identify attack patterns and track human entities' behaviors will provide robust defense mechanism with attack prediction capabilities. Our plan is to evaluate the following popular machine learning models such as support vector machine, Naive Bayes, Ensemble learning, and Decision tree and apply in our framework to detect malicious collaborating entities [62].

B. *Coordinated Intrusion Prevention Using Cryptographic Primitives*

To overcome the threats in CPHS frameworks, a hash function based defense mechanism is designed to generate CPHS entity behavioral proofs that contain information from both data traffic and forwarding paths [18, 21]. We measure and evaluate the impact on parameters such as throughput of application, resources depletion, detection and mitigation capability, and extent of system unavailability with the working of the system under attacks.

C. *Co-ordinated Intrusion Detection of Malicious Collaborating Entities in CPHS*

In this step, Threat Index, TI for an IoT node is calculated by the detection framework from the attack sensitive parameters, $x_n^{'}(t)$ using machine learning techniques.

TI indicates the vulnerability of an IoT node based CPH environment to threats and attacks. This threat index is calculated based on the parameters collected from the IoT-based CPH environment. By calculating threat index, performance trend of IoT-based CPH environment from the security perspective, can be identified and communicated to the user or autonomic intrusion response /tolerance can be applied. TI can be calculated over a specified period of time and that can be compared with the benchmark index thresholds obtained with the help of historical training and machine learning. Machine learning is performed by the collection of data, with and without attacks, with and without control over a long period of time [28]. The comparison of the index threshold with the threat index helps the IoT-based CPH system to gain knowledge of the current state of the environment. To detect the intrusion detection, the computed TI(t) is compared with the threshold values of the Threat Index TI'. The TI thresholds (TI') are obtained with the help of the training dataset where the state of each record is labeled. This step in the proposed framework will address the following questions: (1) How to efficiently integrate information from multiple entities? (2) How to develop attack detection mechanisms that are robust against noises in the detection capabilities of the mechanisms? (3) How to determine the tradeoff between the detection frequency and information?

To address some of the above questions, in this step, data mining models and machine learning (ML) models are generated to understand the relevance of the insights of the attack data. Data records collected from simulation environment with and without attack are used as training dataset for identifying the Threat Index thresholds. The training data is derived from the IoT-based CPH is used in the identification of significant parameters and the thresholds of these parameters and the threat index. If the computed TI(t) of a node is greater than or equal to vulnerable state threshold reference TI', the node is identified to be under threat.

D. Adaptive Coordinated Intrusion Response

This research will involve developing a framework for identifying intruders, designing robust and efficient coordinated defense mechanisms with IDM, Machine Learning (ML), and evaluating those mechanisms. The objective is to develop and apply autonomic or self-adaptive techniques to implement adaptive coordinated intrusion response in CPHS.

In this step, upon detecting a node that is under threat, the neighboring nodes are subjected to the response and protection algorithm. This algorithm identifies the intruder and sends the control signal u(t) to isolate the intruder from the IoT-based CPH. The control signal u(t) varies depending upon the type of the intrusion. This control signal reconfigures the IoT and modifies $f(x_1^{'}(t+1), x_2^{'}(t+1),\ldots x_n^{'}(t+1))$ such that TI(t+1) reaches the steady normal state. It should however be noted that $f(x_1^{'}(t+1), x_2^{'}(t+1),\ldots x_n^{'}(t+1))$ also depend on new attack k(t+1).

E. Autonomic Intrusion Tolerance Using Byzantine Fault-tolerant Replication

In this step, autonomic intrusion tolerance is developed that will control replication due to attacks. This steps aims to develop and apply autonomic or self-adaptive techniques to implement intrusion tolerant CPHS. Byzantine replication approach is applied to construct more robust systems, capable of tolerating the arbitrary behavior generated by an intrusion or by an internal attack. Due to unpredictable behavior of intruders, especially humans, the use of an autonomic loop is chosen among different strategies to reconfigure the system or to switch it to a new operating mode. This will guarantee reliability, security and performance requirements.

Ideally, attacks to IoT-enabled CPHS should be prevented or detected in a timely manner so as the attacker is kept out of the system. However, in some circumstances, attacks may never be completely prevented and some of them may not even be detected, and this is a great challenge for some applications that require the continuation of their services even when the system is under attack or when some of its components have been compromised by an attacker or intruder.

Byzantine-resilient state machine replication is a powerful abstraction that has been widely used to implement systems capable of tolerating arbitrary component failures. That is, in such byzantine-resilient systems, n – t replicated state machines maintain their state consistent despite the action of up to t arbitrarily or byzantine faulty state machines. This notion was extended for tolerating malicious attacks or intrusions so that a system operates correctly even when some of its components get compromised by a malicious intruder or attacker. By combining replica diversity, voting and cryptographic schemes, a byzantine state machine replication based intrusion tolerant system (ITS) can mask a number of compromised replicas, so the system can continue operation without a disruption - with perhaps a degraded performance [43,48, 49, 51, 52, 53].

In order to maintain consistent states, replicated state machines must agree on a total order sequence of client operations [34], and many related protocols were motivated by the need to circumvent the consensus impossibility result in asynchronous distributed systems [35], either by adding system model assumptions [36, 46] or by weakening the required consensus properties [50]. Hence, existing protocols for byzantine state-machine replication works on a variety of system model assumptions and protocol properties. Though the ideas supporting byzantine state machine replication had been around for a few decades [5, 34], it took some time until new protocols overcame the prohibitive performance costs of the first solutions, improving byzantine replication performance in aspects like operations latency, throughput, message overhead and minimum number of required replicas [6, 37,38, 39, 7, 44, 46]. These optimized solutions have diverse characteristics, and each one represents a distinct tread-off in terms of cost and efficiency where optimizing one feature usually implies in sacrificing another [48]. For instance, an implementation that optimizes agreement, by using a centralized message ordering scheme, may expose the replication protocol to specific adversary attacks [40]. Other protocols that separate the ordering phase from the execution phase to improve resource utilization [42], as fewer replicas area required, are more vulnerable to certain attacks on a specific set of replicas (ordering replicas). Solutions that use specially equipped components, such as trusted components [46], are more efficient in resource and time, but have the whole robustness and cost of the system dependent on the implementation of such trusted components, and so forth.

Though these optimized protocols can also be applied for intrusion tolerance [43, 46, 51], one of the main concern is the conventional non-correlated faulty assumption because an intruder can explore vulnerabilities of a compromised replica to attach others. Thus, replica diversity, intrusion detection, reconfiguration and proactive recovery are techniques habitually combined in such settings to implement more robust byzantine state machine replication [41, 43, 46, 48, 51].

As previously mentioned, because byzantine state machine replication based ITS has additional security concerns when compared to conventional byzantine replication, such as the existence of an intelligent adversary and the need for confidentiality - not only integrity and availability -, performance issues already addressed for fault-tolerant systems become more vital. Attacks to slowdown the system or even denial-of-service attacks are of great concerns [61]. For example, attackers may exploit specific implementation characteristics to slowdown execution in a way that makes the system unusable [40, 45, 47].

In general terms, system robustness depends on a given byzantine replication implementation, level of replica diversity, specific attack detection and reconfiguration mechanisms, recovery strategy and so on. These characteristics may come with increasing complexity and cost, with consequences in performance. Moreover, if such defense mechanisms are trigged too often or if system configuration is not adequate, system performance may be affected to an unacceptable level. Additionally, CHPSs do not always operate in controlled environments, so unexpected conditions may occur during execution in physical processes as well as in the network.

To handle such dynamicity, adaptive timeouts [40] have been used to avoid the exploitation of long timeouts that delay protocols steps. A version of byzantine replication has also been developed, where the batch size and batching timeout are regulated by a controller so as to optimize message throughput and delivery time [6, 7]. Other systems have proposed ways to automatically adapt server redundancy to the level of attach alerts [54].

While adaptive mechanisms are suitable to protect systems against malicious and non-malicious variations in client and system activities, such adaptive mechanisms may also need to be modified to cope with unanticipated changes in the computing environment and/or client/system requirements - such as new system/network configurations, new system component versions or even new SLAs. Moreover, changes in defense policies may be required: responses to attacks could trigger new modes of operation aiming at, for example, a degraded but safer operation - following dynamically modified policies. Thus, adaptation must be on-the-fly and without complete previous knowledge or anticipation of what may occur. To address these issues the system must be equipped with self-manageable or autonomic behavior [8]. Therefore, feedback loops are required for sensing both the environment and system requirements, and systems should dynamically adapt themselves according to perceived environment and higher level policies.

Few approaches in the literature have successfully addressed some autonomic properties for intrusion tolerance [52], but developing fully autonomic byzantine-resilience state machine replication remains a challenge. We have effectively applied autonomic or self-adaptive mechanisms to implement some fault tolerant distributed protocols, such as failure detection [9] and group communication [10]. In these autonomic approaches, the protocol is itself an object to be managed by a built-in controller. To design an autonomic byzantine state machine replication protocol, a number of questions must be further addressed, among others: What should be the protocol performance objectives and how to express them? How general defense polities can be translated into protocol objectives? How system and protocol dynamics can be modelled in the loop? When and how to adapt to distinct modes of operations and distinct optimized versions of byzantine replication? How frequently should the system components and protocol variables be monitored and reconfigured?

As shown in figure 2, the "Adaptive Coordinated Intrusion Response" module will produce dynamically defined objectives for the Byzantine Fault-Tolerant Replication (BFT) configuration, according to security alerts, intrusion detections, new modes of operation, resource conditions, and so on. Such objectives have to be compared against perceived behavior from the BFT sensors, and the calculated error or deviation is passed to the BFT Controller that will apply the appropriated control laws to produce new setups that will adjust the BFT protocol to the intended behavior.

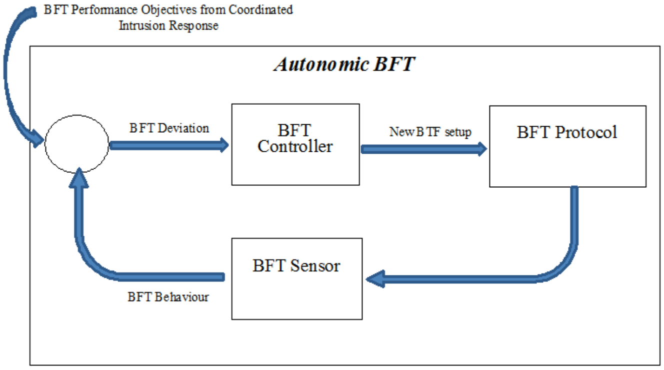

Fig.2 Autonomic Intrusion Tolerance using Byzantine Fault Tolerant Replication

IV. Threat Modeling With Human Entities

Nearly 95% of all the security incidents are caused by human errors. The dynamics and cyber-physical entities of CPHS environment can be influenced by human entities. Human entities add uncertainty to the security of the CPH systems since they are prone to cause intentional (malicious intent of compromising the system) or unintentional (common innocent mistakes made without any malicious intent) errors. The attackers can take advantage of this weakness in the system and deploy sophisticated attacks. Without enough security measures and monitoring, human entities' identity can easily be compromised and stolen, and replicated to produce collaborative attacks on CPHS. With the identity compromise, the privacy of human actors is effectively breached. In order to model threats based on human entities involved in CPHS, we need to categorize the types of security risks for CPHS by identity compromises and types of privacy risks for human entities in CPHS environment [19, 22].

To defeat collaborative attacks based on human errors, we need a systematic way to coordinate defensive mechanisms. Defenses that are not intentionally coordinated may exhibit exactly the same behavior as attacks. We need to discover (1) Can coordinated defensive actions with IDM, when launched together, can be much more effective that defenses that are in effect but not coordinated. (2) Can coordinated defenses have superior defending power (3) Can defensive actions that are not properly coordinated have less damage than ordinary defense and deteriorate the performance of other defense mechanisms. The current problems in coordinated defense with IDM include (1) detection of normal individual attacks, (2) detection and classification of collaborative attacks that consist of individual attacks, (3) how to use identity management to identify intruders and security breaches, (4) how to assign computation power to defense mechanisms and entities, (5) how to properly assign priority to attacks and how to identify resources in the dynamic and fast-changing environment so as to ignore small attacks and concentrate on large attacks, and (6) how to build a model of coordinated defense with Identity Management (IDM).

Modeling Attacks Using Causal Relationships

An attack can consist of combination of several human errors (intentional errors or unintentional errors) that follows one another. For example, an attack will have to first breach perimeter security of a system (e_1) and stole the identity of human entity to gain access to certain level of CPHS and be able to install a virus (e_2). Then the attacker may install further software to propagate this attack to other hosts (e_3). In this paper, we will focus on intentional errors because unintentional errors can be monitored through a central monitor and caught. But intentional errors are hard to identify and defend against, especially if it is a collaborative attack.

In the *causal model*, a state of an individual attack caused by a sequence of intentional human errors represents finite period of individual attack execution. Unlike the state of a variable, the state here refers to a stage in the execution of the attack. A malicious human entity can initiate communication with other collaborating human entities with malicious intent. The sending and receiving of messages must be modeled as state transitions. Examples of legal operations in a state include subversions of individual operating system, individual network port scans, etc. An event causes an individual attack to change its state. We can categorize the attacks by intentional human errors into two types of attack events, namely collaborative attack events and individual attack events. For example, sending or receiving a message constitutes an event. Individual attack events indicate transitions between individual attack states. Since each event is associated with a state transition of an individual attack and vice versa, the set of events for an individual attack can be viewed as the state transition function of the attack.

Analyzing attacks with this modelling where events are partially ordered using a happens-before relationship leads us to several interesting insights when we consider collaboration among attacks. When two attacks collaborate, it may lead to various different attack outcomes. We identify two distinct cases called "positive" and "negative" collaboration. Positive happens when two independent attacks collaborate to increase the number and effects of the resultant damage events. One simple example is where one attack communicates the breach of a host to another attack. At this point the other attack may utilize the same entry point to mount its own attacks on the system. Further this may lead to some additional attack actions on the system as well.

There are cases where the collaboration or interleaving of two attacks may result in nullifying each other or reducing the final damage. This is in essence one attack interfering with another attack known as negative collaboration. For example, in the case of CPHS, mounting a denial of message or flooding attack in the presence of a wormhole attack may not result in the network resources being exhausted. The excessive communication will cause drop in bandwidth available and will interfere with the high-speed tunnel that is needed by the wormhole attack.

We employ causal graph to map the attack patterns through human errors. *A causal graph* G=<V, E> for a set of causal rules of an attack is a labeled digraph with vertices V={e| events} and edges E={<p, q> | there exists a causal relationship c, local operation L, and predicate B such that <p, c, q, L, B> is a causal model}. The vertices and the edges are labeled with their corresponding events and causal relationships. By identifying all attack events that occur during individual and collaborative attacks and establishing a partial order (or causal relationships) among all attack events we can produce a 'causal attack graph'. We can verify the security properties of the causal attack graph using model-checking techniques. Specifically, we can find the existence of a sequence of events that lets the security checker to proceed from the initial state to the goal state. The causal model can help us in modeling large scale networks, where it can model attacks that are sequential as well as concurrent. The causal model can model coordination of entities by exchange of messages. In case the pre-conditions and post-conditions of attacks that satisfy change dynamically, the causal model can capture the change that the state-of-art attack graph reduction techniques cannot. The causal model can describe timing of attacks. Attacks may need to be operating within a specific time interval and traditional attack graph analysis did not consider it. The casual model can represent unsuccessful attacks. Some attempted attacks are never successful and cannot be modeled by traditional attack graphs [56].

V. Conclusion

There are several security issues in the IoT-based CPS. Human participation in CPHS deepens those security issues due to the vulnerabilities in the technologies and the human involvement in the CPHS environment. In this paper, we have proposed a holistic security framework for security threats in the IoT-based CPHS environment. In addition, we have also developed threat modeling involving human elements in the CPHS environment. We believe that research questions and directions that we have identified around the framework for the security in CPHS are worthwhile to pursue.

References

[1] G. Schirner, D. Erdogmus, K. Chowdhury and T. Padir, The Future of Human-in-the-Loop Cyber-Physical Systems, in IEEE Computer, vol. 46, no. 1, pp. 36-45, 2013.

[2] J. Fraga, and D. Powell, A Fault- And Intrusion-Tolerant File System, in proceedings of IFIP 3rd International Conference on Computer Security, Dublin, Ireland, pp. 203-218, 1985.

[3] S. Hossain, S. Etigowni, K. Davis, and S. Zonouz, Towards cyber-physical intrusion tolerance, in proceedings of the IEEE International Conference on Smart Grid Communications (SmartGridComm), Miami, FL, USA, pp. 139-144, 2015.

[4] S. Sowe, E. Simmon, K. Zettsu, F. de Vaulx and I. Bojanova. Cyber-Physical-Human Systems: Putting People in the Loop, in IT Professional, vol. 18, no. 1, pp. 10-13, Jan.-Feb. 2016.

[5] L. Lamport, R. Shostak, and M. Pease. The Byzantine Generals Problem, in ACM Transactions on Programming Languages and Systems. vol. 4, no. 3, pp. 382-401, Jan.-Feb. 2016.

[6] M. Castro and B. Liskov, Practical Byzantine fault-tolerance and proactive recovery, in ACM Transactions on Computer Systems (TOCS), vol. 20, no. 4, November 2002,

[7] A. Sá, A.Freitas, and R.Macêdo, Adaptive request batching for byzantine replication, in Operating Systems Review, vol. 47, no. 1., pp. 35-42, 2013

[8] J. Kephart and D. Chess, The vision of autonomic computing, in IEEE Computer, vol. 36, no. 1, pp. 41-50, 2003.

[9] A. Santos Sá, R, José, and A. Macêdo. QoS Self-configuring Failure Detectors for Distributed Systems. Frank Eliassen;; Rüdiger Kapitza. Distributed Applications and Interoperable Systems, 6115, Springer Lecture Notes in Computer Science, pp.126-140, 2010.

[10] R. Macêdo, A. Freitas, and A. Sá. Enhancing group communication with self-manageable behavior. Journal of Parallel and Distributed Computing. vol. 73, no. 4. pp. 420-433, April 2013

[11] R. Dickerson, E. Gorlin, and J. Stankovic. Empath: a continuous remote emotional health monitoring system for depressive illness, in Proceedings of the 2nd Conf. on Wireless Health San Diego, CA, 2011.

[12] S. Andrade, and R. Macêdo, Architectural design spaces for feedback control concerns in self-adaptive systems. in Proceedings of the 25th International Conference on Software Engineering and Knowledge Engineering (SEKE), Boston, USA, pp. 741-746, June, 2013

[13] S. Andrade, and R. Macêdo. A non-intrusive component-based approach for deploying unanticipated self-management behavior, in Proceedings of ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS '09), pp. 152-161, June 2009.

[14] E. Lee. Cyber-Physical Systems: Design challenges, in Proceedings of the 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC'2008), Orlando, FL, pp. 363-369, 2008.

[15] Y. Deswarte, L. Blain, J., and C. Fabre, Intrusion tolerance in distributed computing systems, In proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy. pp. 110-121, 1991.

[16] M. Huebscher, and J. A. Mccann, A survey of autonomic computing—degrees, models, and applications. ACM Computing Surveys (CSUR), vol. 40, no. 3, p. 7, 2008.

[17] I. Atzori, A. Iera, and G. Morabito, The internet of things: A survey, in Computer networks, vol. 54, no. 15, pp. 2787-2805, 2010.

[18] P. Angin, et al., An entity-centric approach for privacy and identity management in cloud computing. in Proceedings of 29th IEEE Symposium on Reliable Distributed Systems, 2010.

[19] G. Mani, Clone attack detection and data loss prevention in mobile ad hoc networks. in International Journal of Space-Based and Situated Computing, vol, 5, no. 1, pp.9-22. 2015

[20] G. Mani et. al., Internet of things as a methodological concept, in Computing for Geospatial Research and Application (COM. Geo), vol. 201, pp. 48-55, 2013.

[21] W. Wang, B. Bhargava and M. Linderman, Defending against Collaborative Packet Drop Attacks on MANET., in Proceedings of IEEE Symposium on Reliable Distributed System, 2009.

[22] A. Bhargav-Spantzel, et. al. Platform capability based identity management for scalable and secure cloud service access, in Proceedings of IEEE Globecom Workshops. 2012.

[23] S. Kumar**,** T. Vealey and H. Srivastava, Security in Internet of Things: Challenges, Solutions and Future Directions, in Proceedings of 49th IEEE Hawaii Intl Conf on System Sciences (HICSS), Kauai, HI, 2016.

[24] A. Sardana and S. Horrow, Identity management framework for cloud based internet of things, Proceedings of the First Intl. Conference on Security of Internet of Things, pp. 200-203, 2012.

[25] W. Zhao, and Wang. Security Challenges for the Intelligent Transportation System, Proceedings of the First Intl Conference on Security of Internet of Things, pp. 107-115, 2012.

[26] H. Ning and H. Liu, and L. T. Yang, Cyberentity Security in the Internet of Things, vol. 46, no. 4, pp. 46-53,. April 2013.

[27] H. Abie, and Balasingham, Risk-Based Adaptive Security for Smart IoT in eHealth, 2011 Proceedings of the 7th International Conference on Body Area Networks, pp. 269- 275, 2011

[28] P. de Leusse., P. Periorellis., T. Dimitrakos., and S. K., Nair., Self-Managed Security Cell, a security model for the Internet of Things and Services, First Intl Conference on Advances in Future Internet, pp. 47 – 52, 2009.

[29] R. M, Savola. et al., Towards Metrics-Driven Adaptive Security Management in E-Health IoT Applications, in proceedings of the 7th International Conference on Body Area Networks, pp. 276- 281, 2012

[30] X. Xiaohui, Study on Security Problems and Key Technologies of The Internet of Things, im proceedings of Fifth International Conference on Computational and Information Sciences (ICCIS), pp. 407 – 410, 2013

[31] Kozlo et al., Security and Privacy Threats in IoT Architectures, in proceedings of the 7th International Conference on Body Area Networks pp. 256-262, 2012

[32] S. Cirani, G.Ferrari, and L.Veltri, Enforcing Security Mechanisms in the IP-Based Internet of Things: An Algorithmic Overview, Algorithms 2013, 6, 197-226

[33] K.Sonar and H. Upadhyay, A Survey: DDOS Attack on Internet of Things , Intl. Journal of Engineering Research and Development, vol. 10, no. 11, pp. 58-63

[34] F. B. Schneider, Implementing fault-tolerant services using the state machine approach: A tutorial, ACM Computing Surveys, vol. 22, pp. 299–319, Dec. 1990

[35] M. J. Fischer, N. A. Lynch, and M. S. Paterson. 1985. Impossibility of distributed consensus with one faulty process. *J. ACM* 32, 2 (April 1985), 374-382.

[36] C. Dwork, N. Lynch, and L. Stockmeyer. 1988. Consensus in the presence of partial synchrony. *J. ACM* 35, 2 (April 1988), 288-323.

[37] R. Kotla, L. Alvisi, M. Dahlin, A. Clement, and E. Wong, Zyzzyva: speculative byzantine fault tolerance, in Proc. of 21th ACM SIGOPS Symp. on Operating systems principles. ACM, 2007, pp. 45–58.

[38] B. Wester, J. Cowling, E. Nightingale, P. Chen, J. Flinn, and B. Liskov, Tolerating latency in replicated state machines through client speculation, in Proc. of the 6th USENIX Symp. on Networked systems design and implementation (NSDI). USENIX, April 2009, pp. 245–260.

[39] R. Guerraoui, N. Knezeviˇc,´ V. Quema,´ and M. Vukolic,´ The next 700 BFT protocols, in Proc. of the 5th European Conf. on Computer systems (EuroSys). ACM, April 2010, pp. 363–376.

[40] Y. Amir, B. Coan, J. Kirsch and J. Lane, Byzantine replication under attack, 2008 IEEE Intl Conference on Dependable Systems and Networks (DSN), Anchorage, AK, 2008, pp. 197-206.

[41] D. Malkhi and M. Reiter, Unreliable intrusion detection in distributed computations, in Proc. of the 10th Computer Security Foundations Workshop (CSFW). IEEE CS, June 2002, pp. 116–124.

[42] J. Yin, J. Martin, A.Venkataramani, L. Alvisi, and M. Dahlin. Separating agreement from execution for byzantine fault tolerant services. SIGOPS Oper. Syst. Rev. 37, 5 (October 2003), 253-267.

[43] Q. Nguyen and A. Sood, A comparison of intrusion-tolerant system architectures, IEEE Security & Privacy, vol. 9, no. 2, pp. 18–25, Mar./Apr. 2003.

[44] T. Wood, R. Singh, A. Venkataramani, P. Shenoy, and E. Cecchet, ZZ and the art of practical BFT execution, in Proceedings of the 6th ACM SIGOPS/EuroSys European Systems Conference EuroSys'11, Apr. 2011.

[45] P. Sousa, N. F. Neves, and P. Veríssimo, How resilient are Distributed fault/intrusion-tolerant systems, in Proceedings of Dependable Systems and Networks – DSN 05, Jun. 2005, pp. 98–107.

[46] P. Sousa, A. N. Bessani, M. Correia, N. F. Neves, and P. Verissimo, Hfmaighly available intrusion-tolerant services with proactive-reactive recovery, IEEE Transactions on Parallel and Distributed Systems, vol. 21, no. 4, pp. 452–465, 2010.

[47] A. Clement, E. Wong, L. Alvisi, M. Dahlin, and M. Marchetti, Making Byzantine fault tolerant systems tolerate Byzantine faults, in Proceedings of the 6th USENIX Symposium on Networked Systems Design & Implementation, Apr. 2009.

[48] F. Wang, R. Uppalli and C. Killian, Analysis of techniques for building intrusion tolerant server systems, IEEE Military Communications Conference, 2003. MILCOM 2003., 2003, pp. 729-734 Vol.2.

[49] S. Heo, P. Kim, Y. Shin, J. Lim, D. Koo, Y. Kim, O. Kwon, and H. Yoon. A Survey on Intrusion-Tolerant System JCSE, vol. 7, no. 4, pp.242-250, 2013.

[50] M. O. Rabin, Randomized Byzantine generals, in Proc. 24th IEEE Symposium on Foundations of Computer Science, pp. 403–409, 1983,

[51] B. Foo et al.,. Intrusion Response Systems: A Survey: Book chapter in "Information Assurance: Dependability and Security in Networked Systems", pp. 377-416, Morgan Kaufmann Publishers. 2007.

[52] E. Yuan, N. Esfahani, and S. Malek. 2014. A Systematic Survey of Self-Protecting Software Systems. ACM Trans. Auton. Adapt. Syst. 8, 4, Article 17 (January 2014), 41 pages.

[53] A. Valdes et a., An Architecture for an Adaptive Intrusion-Tolerant Server. Volume 2845, Lecture Notes in Computer Science pp 158-178.

[54] K. Goseva-Popstojanova *et al.*, "Characterizing intrusion tolerant systems using a state transition model," DARPA Information Survivability Conference & Exposition II, 2001. DISCEX '01. Proceedings, Anaheim, CA, 2001, pp. 211-221 vol.2.

[55] B.Bhargava, Y. Zhang, N. C. Idika, L. Lilien, and M. Azarmi: Collaborative attacks in WiMAX networks:, Security and Communication Networks 2(5): 373-391 (2009).

[56] N. Idika, and B. Bhargava. Extending Attack Graph-Based Security Metrics and Aggregating Their Application IEEE Transaction on Dependable and Secure Computing 9(1): 75-85 2012.

[57] B. Bhargava, P. Angin, R. Ranchal, S. Lingayat. A Distributed Monitoring and Reconfiguration Approach for Adaptive Network Computing, 6th International Workshop on Dependable Network Computing and Mobile Systems (DNCMS 2015) in conjunction with SRDS'15.

[58] Alampalayam, S. P., and S. Srinivasan. Intrusion Recovery Framework for Tactical Mobile Ad hoc Networks. IJCSNS vol. 9, no. 9 (2009).

[59] Kumar, Sathish Alampalayam. Classification and Review of Security Schemes in Mobile Computing. Wireless Sensor Network vol. 2, no. 6 (2010): 419.

[60] Srinivasan, S. Alampalayam, S.P, Intrusion Detection Algorithm for MANET. International Journal of Information Security and Privacy 5, no. 3 (2011): 36-49.

[61] Chelladhurai, Jeeva, Pethuru Raj Chelliah, and Sathish Alampalayam Kumar. Securing Docker Containers from Denial of Service (DoS) Attacks. In Services Computing (SCC), 2016 IEEE International Conference on, pp. 856-859. IEEE, 2016.

[62] Alampalayam, S. P., and Anup Kumar. Predictive security model using data mining., in proceedings of Global Telecommunications Conference, 2004. GLOBECOM'04. IEEE, vol. 4, pp. 2208-2212. IEEE, 2004.