An Automated Adaptive Security Framework for Cyber-Physical Systems

Elias Seid, Oliver Popov and Fredrik Blix
Department of Computer and Systems Sciences, Stockholm University, Sweden

Keywords: Security Engineering, Control Theory, Adaptive Systems, Security Solution, Multiple Failure, Cyber-Physical Systems.

Abstract: The paper promotes the notion that any security solution for cyber-physical systems (CPS) should be adaptive and based on the type of attacks and their frequency. Namely, the solution should monitor its environment continuously to defend itself from a cyber-attack by modifying its defensive mechanism. Moreover, the research provides analyses of situations where the environment changes dynamically over time, requiring the designated adaptation to contemplate and respond adequately to these changes. In particular, it explores applying adaptive model predictive control concepts derived from control theory to develop specific adaptive security solutions. These systems can make decisions by forecasting their future performance for various modes or options of adaptation. Using quantitative information, the software then selects the adaptations that minimise the cost associated with security failures. This is highly significant considering that CPS are engineered systems built from and depend upon the seamless integration of computational algorithms and physical components. Moreover, security breaches are rising, and CPS are challenged by catastrophic damage, resulting in billions of losses making many of today's solutions obsolete. While security agents issue new sets of vulnerability indicators and patches to address security breaches, these changes are continuous processes ad infinitum. A case study on a medical emergency response system illustrates the essential and salient futures of the proposed adaptive security framework for CPS.

1 INTRODUCTION

Most of today's software systems operate as components of cyber-physical systems (CPSs), which also include physical (e.g., robotic, mobile) components and social components (e.g., humans, enterprise units, and business processes). The components of such CPSs act autonomously by nature but do coordinate in order to fulfill system requirements. Software systems in diverse areas such as healthcare, government, and financial services are often CPS (Griffor et al., 2017; Boyes et al.,2018)

The scope of smart objects encompasses diverse, dynamic, and adaptable networks commonly known as sensor networks. These networks are comprised of numerous smart devices that are geographically dispersed and can be conveniently attached to physical objects. These devices have the capability to monitor various parameters such as temperature, sound, vibration, pressure, and motion. Additionally, they are able to transmit the collected data to software systems located remotely. For example, an emergency healthcare service is a CPS that includes components of hospitals, emergency healthcare workers, doctors, and nurses that perform business processes for dispatching an ambulance to an accident scene, admitting a patient to a hospital emergency ward, and treating the patient's injuries.

Designing a security solution for CPS is more challenging compared to software systems due to the need to consider not only the properties of individual components (such as sensing, communication, and computing components) but also their interaction with the physical environment (Banerjee et al., 2012). Adversaries have the ability to exploit components of CPS that are exposed to heightened risk and vulnerability. This is due to the fact that these components, such as sensors, operate within an open environment that lacks adequate security measures. As a result, security threats such as unauthorised information disclosure, transmission of falsified data, and violations of authentication and authorization protocols must be carefully considered. For instance, items belonging to the CPS may become accessible to the general

Seid, E., Popov, O. and Blix, F.
An Automated Adaptive Security Framework for Cyber-Physical Systems.
DOI: 10.5220/0012469100003648

In *Proceedings of the 10th International Conference on Information Systems Security and Privacy (ICISSP 2024)*, pages 242-253
ISBN: 978-989-758-683-5; ISSN: 2184-4356

public, such as when they are affixed to vehicles or containers. Consequently, the data associated with these items can be readily manipulated, removed, or even destroyed.Sensors have the potential to offer vital security information to the application in real-time, thereby introducing new security requirements that must be acknowledged and fulfilled (Mu¨ller et al., 2016). Dealing with security issues for such complex systems remains a challenging unsolved problem, as CPSs span three realms: physical, cyber (software), and social (Shafi, Q. et al.,2012, ; Morais et al., 2013) each of which comes with its own complexities and vulnerabilities, and they need coordinated security solutions.

Adaptive CPS: are expected to function within environments characterised by significant dynamism and successfully achieve a multitude of goals. When a failure is detected, specifically when a goal is not achieved, as a result of external disturbances such as excessive traffic or unpredictable user behaviour, a new configuration is implemented. However, formulating an approach to mitigate the impacts of changes poses a formidable challenge(K. Angelopoulos et al., 2014)

The primary challenge lies in the unwanted intervention of numerous parameters from the configuration space, each with its own set of goals. As a result, the execution of adaptation mechanism may potentially result in the restoration of security goal satisfaction, but it also carries the risk of failure or exacerbation of security goals.

Control Theory: has emerged as a significant field of study, offering valuable theoretical and practical frameworks for managing complex systems characterised by multiple parameters (inputs) and multiple objectives (outputs). The existing methods for creating self-adaptive software either address each objective separately without considering potential conflicts (Y. Brun et al., 2009; A. Filieri et al., 2011 ; A. Filieri et al., 2014; V. Souza et al., 2012), or they only reactively adapt after a failure has occurred without any consideration for future occurrences (A. Filieri et al., 2015; S. Cheng et al., 2006; C. Klein et al., 2014; K. Angelopoulos et al., 2014 ; P. Zoghi et al., 2014) or proactive measures for failure anticipation. In situations where the workload increases and a specific objective is not achieved, supplementary resources are allocated to the system in order to address the failure.

The existing approaches(H. Ghanbari et al., 2014; D. Kusic et al., 2008; Q. Zhang et al., 2012) address the challenge of managing conflicting demands and the associated costs of adaptation by employing predictive models of the system's environment. These models enable the anticipation of failures and the development of reconfiguration plans that optimise the system's utility output over time. This optimisation is achieved through the application of a control theoretic technique known as Model Predictive Control (MPC) and its various adaptations (J. Maciejowski et al., 2002).

Nevertheless, these methodologies are limited to addressing resource allocation and architectural design, disregarding the aspects of requirements and behaviour within the adaptation space (K. Angelopoulos et al., 2015). Furthermore, the absence of a cyber security methodology that establishes a connection between the components of Model Predictive Control (MPC) and those of adaptive software system hinders designers from employing this technique in domains beyond service-based applications. Currently, the existing approaches primarily concentrate on service-based applications. Several studies (Gaggero and Caviglione, 2016; Ghanbari et al., 2014; Kusic et al., 2009; Roy et al., 2011) have employed predictive control techniques in the field of cloud computing to ensure the fulfilment of nonfunctional properties. Based on current understanding, cyber security have displayed hesitancy in embracing predictive control for various domains due to the absence of adequate tools and methodologies for modelling security goals and parameters within a control-theoretic framework.

This paper offers an analysis of the integration of principles derived from the disciplines of Cybersecurity and Control Theory, building upon our previous research on security attack event monitoring (Seid et al. 2023). The main aim of this study is to present an adaptive security solution through the use of key components of model predictive control within the domain of CPS.

The purpose of this study is to determine a relationship between the aforementioned components and the security goal of the system being examined. Moreover, the current analysis presents XA4AS [1] framework that facilitates the formulation of the necessary analytical models for model predictive control and adaptive security solutions for CPS. The framework that has been developed incorporates fundamental concepts derived from the discipline of Software Engineering(K. Angelopoulos et al., 2014;)

The implementation of model predictive control ensures the reduction of excessive overshooting, effective handling of constraints, and the achievement of an optimal balance among competing security objectives over a specified time frame, facilitated by the utilisation of prioritisation techniques. This study em-

[1]Extended Asfalia (Framework) for Adaptive Security (of Cyber-Physical Systems)

ploys the Analytic Hierarchy Process (AHP) methodology, which has been previously documented in scholarly literature(J. Karlsson).

We employ a combination of model predictive control techniques, as described in (M. Gagger et a., 2015; A. Filieri et al., 2014; A. Filieri et al., 2015), to forecast the future behaviour of the controlled system within a defined time period. Additionally, we propose the adoption of a dynamic methodology for generating adaptation strategies aimed at reducing the deviation of each security mechanism from the predetermined threshold set by the relevant stakeholders.

The remaining parts of this paper are structured as follows. Section 2 provides the research foundation for our work, while Section 3 illustrates the formalisation process, and Section 4 offers a case study. Section 5 introduces the XA4AS framework and experiment, while section 6 focuses on the behavioural model. Section 7 provides discussions, while Section 8 concludes and addresses future work.

2 RESEARCH BASELINE

The proposed approach incorporates principles and methodologies derived from Software engineering, Cyber security and systems engineering disciplines. This section provides a brief summary of the key elements within each of these domains that were used as the foundation for our research work.

2.1 Security Attack Event Monitoring for Cyber Physical-Systems

This section presents the Asfalia framework that supports the monitoring of security attack events for CPSs, and the framework spans the three realms of a CPS. Moreover, the framework supports cross-realm analysis and monitoring, which spins off security events across realms. Our models focus on realm-specific adversaries, meaning that they span the three realms of a CPS (cyber, physical infrastructure, and social). We also analysed the interdependent relationships among realm-specific attack models. The AM depends on the VM model in revealing realm-specific vulnerabilities, and vulnerabilities captured by (the VM) spin off and provides inputs to the next realm (AM). Thus, a suitable attack mechanism is selected by taking advantage of the weaknesses of the VM. More detailed information can be found in (Seid et al., 2023). The Asfalia analysis process consists of the following steps.

Vulnerability Model(VM): This model captures the attack patterns, potential threats, and type of asset, in which an asset is a potential target for cyber attacks.

Attack-Mechanism Model (AM): This model captures design strategies for different attack pattern mechanisms. More importantly, it builds attack mechanisms by employing goal models, domain assumptions, attack mechanisms, and task operationalization artefacts.

Behavioural Model (BM): Building a complete behavioural model for very complicated systems such as CPS, with many complex and heterogeneous states, is often challenging.

Event Model (EM): This model captures events derived from Behavioural Models (BM).

2.2 Security Models

Monitoring and assessing requirements satisfaction is essential for self-adaptive systems. Awareness Requirements (AwReqs) are used to track the success of other requirements, inspired by feedback control theory. An AwReq sets a constraint that prompts adaptation upon violation and Each AwReq has variables called indicators that measure the success of a monitored requirement. Two types of parameters affect indicator values: environmental parameters that cannot be controlled, and control parameters that can be adjusted at runtime (K. Angelopoulos et al., 2014). We adopt some of their concepts into our framework to monitor security mechanisms(solutions).

Moreover, another kind of security requirement that has been incorporated into our approach is known as an evolution security goal. These conditions are applicable under specific circumstances and serve as substitutes for other requirements, either temporarily or permanently. These modifications are implemented through a series of actions referred to as EvoSm operations. In addition to the specified requirements for the system under consideration, there are also constraints imposed on the process of adaptation itself, which are referred to as adaptation requirements (AdReqs) (V. Souza et al., 2012). In our proposed framework, we have extnded an AdReqs to generate two distinct security mechanisms, namely EvoSm (evolvable security mechanism) and ASm (adaptive security mechanism), respectively.

2.3 Dynamic System

A sufficiently accurate quantitative dynamic model can be obtained through system identification techniques (L. Ljung et al., 2010) and can be utilised for control design. The dynamic relation between the vector of control parameter values, u(t)$\in$ (R^m) and the

vector of indicators, y(t) ∈ (R^p), is described as follows.

$$y_i^{(t)} = \sum_{j=1}^{p} \sum_{k=1}^{n_y} \alpha_{ijk} y_j^{(t-k)} + \sum_{j=1}^{m} \sum_{k=1}^{n_u} \beta_{i_{j\,kU_j^{(t-k)}}} \quad (1)$$

For all i = 1, . . . , p, and with $\alpha_{(i_{j_k})} \in R$.

The quantitative dynamic model (1) links the indicator yi at time t to past values of all indicators, taking into account potential mutual influences and control parameter values. The adaptation process can be guided by model (1) to capture implicit relationships among indicators. It should be noted that in cases where certain variables do not have an impact on the value of the indicator yi(t), the corresponding parameters are effectively zero.

The discrete-time state-space dynamic model provides a more concise and equivalent representation of this relation.

$$\begin{cases} x(t+1) = A_x(t) + B_u(t) \\ y(t) = C_x(t) \end{cases} \quad (2)$$

The vector denoted as x(·) represents the dynamic state of the model. In the context of physical systems, the state x(·) is commonly linked to tangible physical quantities. However, it is important to note that the state can also be an abstract representation of the system, lacking of direct measurability. The matrices (A, B, C) encompass the complete representation of the relationship between inputs and outputs within the system, and are derived as a result of the System Identification procedure.

The analytical model described by Equation (1) demonstrates that the output of the system may exhibit a correlation with previous outputs and control inputs (V. E. S. Souza et al., 2011). The aforementioned entities are dynamic systems within the field of Control Theory. If there is no apparent connection between the previous behaviour of the indicators and control inputs, matrix A will consist entirely of zero elements. In this scenario, the system merely maps inputs to outputs without any dynamic relation.

$$y(t) = CB_u(t-1)$$

The use of Equation (2) enables the design of a control system that possesses the capability to modify the values of all control parameters. This adjustment is performed with the objective of achieving convergence of each indicator to the value specified by an AwReq threshold. This design assumes that the selected set of control parameters has the capacity to guide the system towards the predetermined goals. In contrast to qualitative adaptation, quantitative models offer the advantage of precise conflict resolution (K. Angelopoulos et al., 2014).

2.4 Model Predictive Control

We present a receding horizon model predictive control (MPC) approach (E.Camacho et al.,2004; J. Maciejowski et al., 2002) that effectively addresses the management of multiple conflicting goals through the use of multiple control parameters. When the controller is enhanced with a Kalman Filter (KF) (L. Ljung et al., 2010), it has the capability to acquire knowledge in real-time and adjust the controller according to the behaviour of the system. This allows the controller to overcome inherent inaccuracies arising from dynamics that are not accounted for in model (2), as well as unknown disturbances affecting the system.

(MPC) is a control technique that employs an optimisation problem to determine a set of control parameters (actuators), denoted as u(·), in order to achieve a desired set of goals, denoted as y∘(·), for a set of indicators, denoted as y(·), over a prediction horizon H. The control parameters u* are determined at each control instant t by minimising a cost function J(t), while adhering to specified constraints. The optimisation problem involves making predictions about the future behaviour of the system using the dynamic model (2).

As a result, a derived solution refers to a planned arrangement of forthcoming control parameter values $U^*= U_t^* + 1, ...U_t^* + H - 1$ across the anticipated time horizon. Effective planning is particularly crucial in situations where there is a delay in the occurrence of changes in control parameters.

The receding horizon principle applies only the first computed value u_t^* to the system, u(t)=u_t^*. Creating perfect models for real-world systems is impossible due to their dynamic behaviour. Therefore, plan corrections are necessary at each step and the horizon is reduced by one unit. The plan may fail due to external disturbances such as system workload changes. In essence, the plan would have been followed if a perfect model and no disturbances were present, which is not feasible. At the next control instant, a new plan is created based on the updated measured values of indicators to overcome this obstacle. This accounts for modelling uncertainties and unanticipated system behaviours (2). The model has been incorporated into our framework for adaptive security strategies and is integrated within our architecture, as detailed in the subsequent section.

$$\overbrace{\begin{bmatrix}\Delta x(t+1)\\ y(t)\end{bmatrix}}^{\tilde{x}(t+1)} = \overbrace{\begin{bmatrix}A & 0_{n\times p}\\ C & I_{p\times p}\end{bmatrix}}^{A} \overbrace{\begin{bmatrix}\Delta x(t)\\ y(t-1)\end{bmatrix}}^{\tilde{x}(t)} + \overbrace{\begin{bmatrix}B\\ 0_{p\times m}\end{bmatrix}}^{B} \Delta u(t)$$

$$y(t) = \overbrace{\begin{bmatrix}C & I_{p\times p}\end{bmatrix}}^{\tilde{C}} \overbrace{\begin{bmatrix}\Delta x(t)\\ y(t-1)\end{bmatrix}}^{\tilde{x}(t)} \qquad (3)$$

Figure 1: Dynamic Model.

$$J_t = \sum_{i=1}^{H} \left[y^{\circ}_{t+i} - y_{t+i}\right]^T Q_i \left[y^{\circ}_{t+i} - y_{t+i}\right] + \left[\Delta u_{t+i-1}\right]^T P_i \left[\Delta u_{t+i-1}\right],$$

Figure 2: Cost Function.

$$\begin{aligned}
\text{minimize}_{\Delta u_{t+i-1}} \quad & J_t \\
\text{subject to} \quad & u_{\min} \le u_{t+i-1} \le u_{\max}, \\
& \Delta u_{\min} \le \Delta u_{t+i-1} \le \Delta u_{\max}, \\
& \tilde{x}_{t+i} = \tilde{A} \cdot \tilde{x}_{t+i-1} + \tilde{B} \cdot \Delta u_{t+i-1}, \\
& y_{t+i-1} = \tilde{C} \cdot \tilde{x}_{t+i-1}, \\
& \quad i = 1, \ldots, H, \\
& x_t = x(t).
\end{aligned}$$

Figure 3: MPC Optimization Problem.

$$\begin{aligned}
\hat{y}(t) &= C\hat{x}(t) \\
\hat{x}(t+1) &= A\hat{x}(t) + Bu(t) + K\left(y(t) - \hat{y}(t)\right)
\end{aligned}$$

Figure 4: KF.

3 FORMALISATION

In order to clarify the core rationale of the model predictive control approach, it is necessary to modify the dynamic model 2.

State variation is $\Delta x(t)$=x(t) - x(t - 1), while control increment is $\Delta u(t)$= u(t) - u(t - 1). The system output y(t) remains unchanged, but now reflects state variations x(t) rather than state values x(t). The new dynamic model (3) predicts over a finite horizon H. The controller will use this to predict the values of states and indicators after H time steps from the current one(). MPC controller reduces cost function as shown in fig 3 below.

Where $Q_i \in R^{p\times p} and P_i \in R^{m\times m}$ are symmetric positive semi-definite weighting matrices are utilised to represent the relative significance of the gap between the goals and the present values, as well as the resistance to change in the actuators' values. Specifically, Qi refers to a matrix with diagonal elements that represent the weights derived from the application of the Analytical Hierarchy Process (AHP) (J. Karlsson et al., 1997). This process involves stakeholders conducting pairwise comparisons to prioritise the goals that have been elicited. This implies that in cases where all goals cannot be achieved simultaneously due to conflicts, the controller will prioritise the fulfilment of goals with higher weights of the other control parameters. Consider the weight matrix Q as $Q := Q_1 = Q_2 = ...Q_H$, and the weight matrix P as $P := P_1 = P_2 = ...P_H$. The resulting optimisation problem for the Model Predictive Control (MPC) can be formulated as follows:

The aforementioned formulation can be considered as a convex Quadratic Programming (QP) problem, as stated in reference (Y. Wang et al., 2010). The issue exhibits a time complexity of $O(H^3m^3)$. A potential solution to the issue entails the formulation of an optimal strategy plan for the future $\Delta u^*_{t+i-1}, i = 1.....,H$, but only the first one is applied, i.e., $\Delta u(t) = \Delta u^*_t$, and the New control signal is: $u(t) = u(t-1) + \Delta u(t)$.

The MPC strategy operates under the assumption that the system's state can be measured. However, in numerous instances, this assumption is not feasible. Indeed, due to the frequent absence of a correlation with physical quantities, the meaningful interpretation of x(t) becomes unattainable, as a result, rendering its measurement impossible. However, according to the dynamic model above, it is feasible to estimate its value by measuring the values of y(t) and u(t).

In order to achieve this objective, we adopt (K. Angelopoulos et al., 2014) and employ a Kalman Filter (KF) algorithm that calculates an estimated value $\hat{x}(t+1) for the state x(t+1)$. This estimation is based on measurements of the applied control signal u(t) and the output y(t) as follows:

KF variables are often referred to as "hat" variables, such as $\hat{x}(k)$ and $\hat{y}(k)$, to differentiate them from dynamic model variables. The KF as shown in fig 4 estimates the output $\hat{y}(t)$ based on the state estimate $\hat{x}(t)$ to measure the difference between the predicted and real values. The Kalman gain, or value of K, adjusts the dynamics of the KF by weighing the difference between the predicted value $\hat{y}(t)$ and the real value y(t) (L. Ljung et al., 1999). Use the estimate $\hat{x}(t)$ instead of x(t) to solve the optimisation problem as shown in Figure 4.

In some cases, the time needed to calculate the next control action value may be longer than the time between two subsequent actions. Utilising the proactive nature of the MPC is another option for meeting real-time deadlines. The MPC calculates a plan of future actions $\Delta u_{t+i-1}, i = 1, ..., H$ at each iteration step. The receding horizon principle dictates that only the first action $\Delta u(t) = \Delta u^*{}_t$ is applied.

If the solver takes longer to converge at the next

control instant, a new control action may be needed to find the optimal solution. The previously computed plan can be stored. To apply the second control action, use $\Delta u(t+1) = \Delta u^*(t+1)$. This approach is suboptimal as it misses the last information about the measured output, but still meets real-time deadlines.

4 THE CASE STUDY: MEDICAL EMERGENCY RESPONSE SYSTEM

A smart item is an Internet of Things device that generates data about itself, an object it is associated with, or its environment. For instance, a sensor that measures the temperature of a physical environment and transmits the data is a smart item. The domain of smart items encompasses heterogeneous, dynamic, and flexible networks commonly referred to as sensor networks (Conway, J. et al., 2016)

The case study Smart Items Medical Emergency Response (MERS) is adopted from the SERENITY project. [2]. The validation of our framework will be demonstrated through the use of MERS, which will be discussed in the subsequent section.

4.1 Implementation of the MERS Security Model

The process begins with a timer event, as depicted in figure five. The event is triggered when the patient feels dizzy and requests assistance. The request is then received at the emergency centre, which in turn checks the availability of a designated doctor and also obtains a list of available social workers. Then If a doctor is available, he/she interrogates the patient's medical data through his/her e-health terminal and analyses them to determine an appropriate treatment. The doctor might even call the patient for further information if required. Furthermore, the doctor then writes an e-prescription and uploads it onto the system for the patient to access it. However, when the designated doctor is not available, MERS generates a list of qualified substitute doctors by obtaining data from a database. MERS sends a message to all substitute doctors who match the qualifications of the designated doctors. The doctors who are available reply to MERS.

MERS selects the doctor who replied first and assigns him/her the task. MERS also provides doctors with medical data from patients in the data repository. Then, the assigned doctor checks the patient's medical data and decides on the appropriate treatment to generate an e-prescription. The patient receives the e-prescription in his/her e-health terminal and decides either to go to the pharmacy and buy the prescribed medicine or to ask MERS to deliver the medicine at home. In the latter case, MERS receives the list of available social workers from the database and selects a suitable worker for the task.

MERS also provides the social worker with authentication to access that patient's e-prescription. After receiving the task assignment, the social worker acknowledges receipt and goes to a pharmacy. The pharmacy authenticates the patient and gives him/her the medication. Finally, the medication is delivered to the patient's home.

4.2 Vulnerability Model(VM)

The vulnerability model used in our case study, (MERS), comes from the vulnerability model component of Asfalia. This model captures the type of attack patterns, potential threats, and the type of asset, with an asset being a potential target for cyber attacks. The VM consists of the following sub-elements.

Threat: This is the potential for abuse of an asset that will cause harm in the context of the problem. Vulnerability: This is a weakness in the system that an attack exploits. Asset: This is anything that has value to an organization, and it can be tangible (physical) or intangible (non-physical) with respect to the target of the attack.

Vulnerability analysis of the application realm: Through this analysis, we captured 8 requirements of the control station application, which are intended to provide medical data Security Goal (G1), obtain patient-specific settings (G2), send abnormal behaviour to MERS (G3), deploy first-aid team (G4), check the patient's recent medical history (G5), write an e-prescription (G6), assign a first-aid team (G7), and send an acknowledgment to MERS (G8). These security goals are decomposed into sub goals and then operationalized with tasks, which assign specific functionality to support the software components, as shown in the entire implemented model [3]

5 XA4AS FRAMEWORK

The methodology we employ consists of two distinct stages: the design time phase and the runtime phase.

[2] http://eu-serenity.sourceforge.net/

[3] https://www.dropbox.com/scl/fi/0v8995uho1vm3857nz3pq/Full-model-of-our-case-study.pdf?rlkey=1sjq4s0o0j4yvof0ha68x32db&dl=0

In the initial stage, the necessary models for the synthesis and tuning of the MPC controller are obtained. As a result, in the subsequent stage, the controller is implemented within our adaptation framework and modifies the control parameters of the target system as necessary.

5.1 Design-Time

Table 1: Security Mechanisms for the Critical Security Goals.

Security Mechanism	Security Goal
Cabling security between PDA and control station server	SG1
Control pharmacy access to doctor e-prescription	SG2
Cabling security between PDA and MERS	SG3
Secure access layer for e-health application	SG4
Secure pipe between MERS and control station application	SG5
Secure pipe between patient mobile and control station application	SG6
Cabling security between localization server and internet	SG7
Control access to MERS	SG8
Access layer for hospital application system	SG9
Secure pipe between hospital application system and e-health system	SG10

Our approach begins with the elicitation of various security goals and security mechanisms related to the target system. When all goals have been refined, ASm are assigned to those that are deemed most important and likely to fail. An ASm specifies a reference goal R i (.) for the output of the controller. Table 6 lists all of the Medical Report Emergency System's reference security goals. The control station application relies on two sub-goals to achieve the goal of providing support to write e-prescriptions (G6): "add a signature to the prescription (G7)" and "receive requests for treatment (G8)." Furthermore, these sub-goals necessitate the inclusion of components in an application that provide the desired functionality.

The control station application must also allow authorised users to deploy a first-aid team (G4). To achieve this goal, it must first support two sub-goals: finding available first aid teams (G11) and authenticating and authorising first aid teams (G9). These sub-goals are eventually met by the functionality that allows access to PDA Task2 or (T2) and allows selecting Task 3 (T3) and deploying First-Aid Team Task (T4) as shown in [4]

During this stage of the design phase, the domain experts, in collaboration with the stakeholders, engage in the analysis and evaluation of relevant conditions. They proceed to specify EvoSm for the system that is being developed. The operations of EvoSm as defined for the ASm of the Medical Report Emergency System.

Table 2: Reference Security Goal.

CRq	Reference
ASm1	R1=80
ASm2	R2=75
ASm3	R3=100
ASm4	R4=90
ASm5	R5=90
ASm6	R6=100
ASm7	R7=80
ASm8	R8=100
ASm9	R9=100
ASm10	R10=90

The G2 system, which involves patient information and communication with MERS, has been enhanced through the implementation of three components: T1, a secure channel connecting MERS and the control station application; T2, a secure access layer for the control station application; and T3, a secure channel connecting the patient's mobile device and the control station application. The potential annotation of G3 may consist of either T2 or T2. In the event that T1 fails to yield the desired outcome, T3 will be implemented. In this scenario, the initial EvoSm operation is initiated, resulting in a modification of the reference goal from 80% to 70%. The occurrence of the second EvoSm operation is initiated in the event of T3 failure, resulting in the restoration of the threshold to its previous value.

T1 and T2 have the potential to undergo multiple iterations in a sequential manner, as they await each other's completion. In a formal manner, the annotations of G2 can be represented as (T1; T2) +ˆ). In this scenario, the reference security goals have been adjusted to a lower level, specifically from 90% to 80% and from 80% to 60%, respectively.In the event that ASm5 and ASm8 experience a failure lasting in excess of three days, the reference security goal will be temporarily relaxed for a duration of one week. When the occurrence of goal G1 failing exceeds three times per week in the context of

[4]https://www.dropbox.com/scl/fi/0v8995uho1vm3857nz3pq/Full-model-of-our-case-study.pdf?rlkey=1sjq4s0o0j4yvof0ha68x32db&dl=0

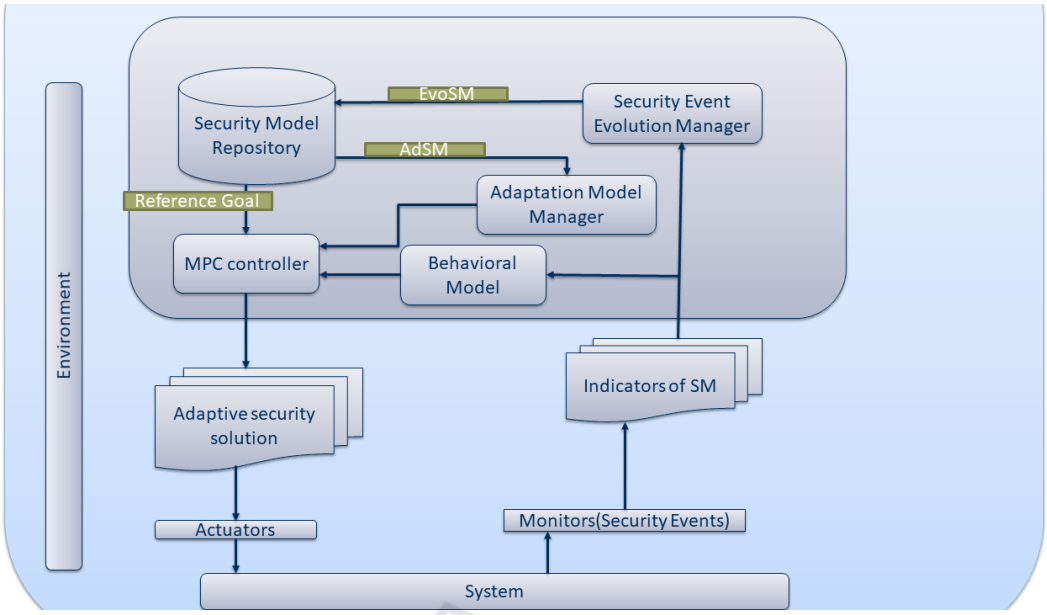

Figure 5: XA4AS.

ASm6, the constraint is consistently modified to four times per week. In the event that ASm7 experiences a failure lasting longer than three consecutive days, the adaptation mechanism will subsequently disregard it for a period of three days. As a result, the Analytic Hierarchy Process (AHP) is employed to determine the weights assigned to each indicator in order to accurately reflect their respective levels of significance. Typically, critical secuirty goals are given higher priority than non-critical ones. The weights correspond to the elements of matrix Q in the cost function. The controller utilises the optimisation function to identify an equilibrium state for each goal, allocating greater resources towards resolving the objectives of higher significance. Regarding the control parameters, their weights are determined through empirical elicitation, with lower weights assigned to the control parameters that require less frequent tuning. The values of matrix P in the cost function are weights.

In our (MRES) case study, it has been observed that the implementation of a secure pipe between the patient mobile device and the control station application (SG6) is a more cost-effective solution compared to establishing a secure pipe between the hospital application system and the e-health system (SG10). Moreover, the former approach also offers the advantage of immediate effectiveness. The priorities for the indicators of MERS and the weights for control parameters were obtained through elicitation, as presented in table 3 and table 4, respectively.

The final set of security goals to be elicited relates to the adaptive security mechanism (ADSM). The

Table 3: SM Indicators.

Sm Indicators	Priority
ISm1	0.20
ISm2	0.4
ISm3	0.4
ISm4	0.08
ISm5	0.2
ISm6	0.05
ISm7	0.3
ISm8	0.18
ISm9	0.16
ISm10	0.5

Table 4: Sm Indicators.

Control Parameter	Weight
SG1SG2	1
SG5SG6	1
SG3SG4	1.4
SG7SG8	0.8
SG9SG10	0.6
SG2SG8	1
SG6SG9	1.3
SG10SG3	1
SG8SG2	1.2
SG1SG7	1

achievement of these goals places limitations on the process of adaptation itself. In the context of Model Predictive Control (MPC), an Adaptive Horizon Determination Strategy (ADSM) is employed to define the receding horizon of the controller. This strategy

determines the time frame within which the adaptation plan should be directed towards in the future. The term "ADSM" could also refer to the extent to which control parameters are permitted to vary.

Ultimately, it is necessary to develop a quantitative model, such as Equation 2. To simulate the MERS system without natural laws, we ran a lengthy simulation with frequent changes in control parameters and recorded input and output. We use Matlab and System Identification toolbox to estimate the analytical model of the system. Although the system cannot be accurately simulated, the model can be improved during deployment by implementing a learning mechanism during runtime as shown in Figure 6.

6 BEHAVIORAL MODEL (RUNTIME)

Developing a comprehensive behavioural model for highly complex systems such as CPS, characterised by numerous complicated and diverse states, presents a significant challenge (Cailliau and van Lamsweerde et al., 2017). In order to comprehend how attackers achieve their objectives by compromising security concerns such as confidentiality, integrity, availability, and accountability, it is necessary to analyse the behaviour of the threat environment within the system and how adversaries can exploit vulnerabilities. Once the design phase has been finalised and the system has been successfully implemented, the XA4AS framework, which is focused on control-based security goals, can be effectively deployed to serve as the mechanism for adaptation. The figure presented as Figure 5 illustrates the five primary components of XA4AS.

Security Model Repository. The repository is responsible for storing all the models generated during the design phase and sending information to other components of the framework upon request.

Security Event Evolution Manager. The purpose of this component is to examine the logs generated by the monitors with the aim of detecting specific conditions that would initiate EvoSM operations. When a security objective is substituted, whether on a permanent or temporary basis, it results in the modification of the security repository.

Adaptation Model Manager. The translation of the AdSM (Adaptive Security Mechanism) component into constraints for the optimisation problem of Equation, as depicted in Figure 3, is performed. These constraints pertain to the maximum permissible reduction or augmentation of a control parameter within a single iteration, as well as the weighting assigned to all indicators and control parameters, as represented by matrices Q and P. **Behavioral Model.** The process of black-box system identification may not always yield models that accurately represent the behaviour of the system. Hence, our framework incorporates a learning component that, through the analysis of implemented modifications and the observed values of indicators resulting from these modifications, iteratively adjusts the control law to accommodate changes in the system's behaviour. This particular component serves as an instantiation of the Kalman Filter, as elaborated upon in the preceding section.

MPC Controller. The particulars of this component have been discussed in the preceding section. In summary, the MPC controller obtains the reference goal "R" for each monitored indicator by requesting the security model repository. The algorithm subsequently computes the distances between each indicator and its corresponding reference goal. It then formulates an adaptation plan that minimises these distances, while considering the priorities assigned to each indicator. The objective of this plan is to restore equilibrium, while adhering to the constraints imposed on the control parameters. The proposed plan entails modifications to the control parameters within a predetermined time frame.

For example in MERS, In the event that ASm5 and ASm8 experience a failure lasting in excess of three days, the reference security goal will be temporarily relaxed for a duration of one week, and The controller generates a new strategy that aims to predict and address potential future failures in a receding horizon manner.

The iterative adaptation process involving XA4AS encompasses the subsequent stages in figure 6

- The monitors gather the measurements of all the indicators of the security mechanism.
- The role of the Security Evolution Manager is to assess the presence of any event that would initiate an EvoSm operation. If such an event is identified, the manager then proceeds to update the evolved security goal within the security Model Repository.
- The Adaptation Model Manager is responsible for supplying the Model Predictive Control (MPC) controller with the necessary weights for the indicators and control parameters, as well as the constraints for the optimisation problem.
- The Behavioural Component of the Model Predictive Control (MPC) algorithm uses recent measurements to generate corrected behaviours for the system.

- The Model Predictive Control (MPC) controller uses the current reference goals obtained from the Security Repository and the corrected model to generate an updated adaptation plan. This plan aims to ensure that each indicator value converges to the reference goal within the prediction horizon.
- The actuators are responsible for implementing the initial phase of the adaptive security solution plan onto the system.

Table 5: EvoSm Operations.

CRq	EvoSm Execution
ASm1	Relax(ASm1,ASm1'_70), Strengthen(ASm1,ASm1'_80)
ASm2	Relax (ASm2, ASm2'_60)
ASm3	Relax (ASm3, ASm3'_90)
ASm4	Relax (ASm4, ASm4'_80)
ASm5	Wait(3 days)
ASm6	Replace (ASm6, ASm6'_3)
ASm7	Wait 1 week days
ASm8	Wait (3 days)
ASm9	Realx (ASm9, ASm9'_90)
ASm10	Realx (ASm10, ASm10'_80)

Table 6: Sm Indicators.

Control Parameter	Weight
SG1SG2	1
SG5SG6	1
SG3SG4	1.4
SG7SG8	0.8
SG9SG10	0.6
SG2SG8	1
SG6SG9	1.3
SG10SG3	1
SG8SG2	1.2
SG1SG7	1

7 DISCUSSION

The experiment used a security model with 1,118 components. We created a vulnerability security model (VM) using the multi-realm security event monitoring methodology from our earlier work. Retrospective security goal modelling was used to identify all potential security events in the model. The inquiry attempted to assign security indicators to these events. In result, the framework was used to analyse a security event involving 160 elements in the Vulnerability Model, including threat targets and assets. The Attack-Mechanism Model included 60 elements, including security goals, tasks, mechanisms, and domain assumptions. Furthermore, the research included 1,125 behavioural annotations and found 38 security events. Only 10% of the reported security incidents were rated critical by the Adaptation manager of XA4AS framework. XA4AS adapts to nonlinear behaviours inherent in the system, an essential feature. The simulator exploits nonlinear interactions between inputs and outputs for more realistic behaviour. In practice, most systems have nonlinear input-output interactions. Thus, an efficient adaptation process must address model defects. In Model Predictive Control (MPC), the Kalman Filter (KF) improves model adjustment during system operation, leading to more accurate predictions.

In the MERS security model scenario, a linear model predicted system behaviour. However, this may not always be applicable. For non-linear systems, tailored models or advanced system identification methods can be used (L. Ljung et al., 2010). Nonlinear model predictive control (MPC) formulations (F.Allgower et al., 2000) can also be used. Our analysis suggests that a bottom-up approach to building the security model within the framework is effective. The distinction lies in task definitions within each area. This allows for the detection of domain-specific attackers. We prioritise threats related to cyber, physical infrastructure, and social aspects of CPS in our models.

Furthermore, we analysed the interdependence of attack models across many domains. The (AM) approach relies on (VM) to identify vulnerabilities unique to a specific domain. Once vulnerabilities are found by (VM), they are used by (AM) to guide the selection of an acceptable attack strategy. The technique of selecting this option involves exploiting known vulnerabilities in the VM. A simulation or historical data is required to create the analytical model required for XA4AS to function properly. This is a major shortcoming of the system. Currently, there are no methods to help system designers simulate a model that produces data close to the actual system, making this difficult. This is because CPS lacks its own methodologies. Our future research is to develop approaches for security engineers and establish guidelines for tuning MPC settings.

8 CONCLUSION

The paper's primary objective is to provide an extension of our previous work in the area of security attack event monitoring, with a specific focus on the Asfalia framework. Another objective is to integrate the concept of MPC into the design of adaptive se-

curity for cyber-physical systems (CPS). Therefore, we introduce a framework termed XA4AS (Extended Asfalia (Framework) for Adaptive Security (of Cyber-Physical Systems). This framework aims to facilitate the creation of fundamental analytical models for the deployment of model predictive control and adaptive security solutions within the domain of CPS.

The aforementioned model demonstrates an elevated degree of accuracy in predicting the behaviour of the system. Thus, XA4AS is capable of effectively responding to environmental oscillations and generating adaptive solutions in a dynamic manner. The framework was evaluated through the implementation of the medical emergency response system.

The evaluation findings demonstrate that the application of control-theoretic principles can produce effective adaptation plans for cyber-physical systems, often surpassing the outcomes achieved through approaches based solely on human experience. One significant advantage of our framework is its ability to support security analysts in their analysis of security events and developing adaptive security solutions for CPS. (VM) is designed to detect and capture both adversaries and vulnerabilities and later proceeds to analyse the target of the attack using a realm-specific methodology.

The Attack Model (AM) is developed by constructing a model based on the adversaries that are specified in the (VM). The behavioural model (BM) provides annotations for the system behaviours of the (VM) and the (AM). In the (EM), events are derived from behavioural models. The aforementioned models are useful inputs for the Security Evolution Manager and Adaptation Manager components of the XA4AS framework. The effectiveness of our approach requires further evaluation through the use of a larger volume of case studies.

REFERENCES

Griffor, E. R., Greer, C., Wollman, D. A., Burns, M. J., et al. (2017). Framework for cyber-physical systems: Volume 1, overview.

Boyes, H., Hallaq, B., Cunningham, J., and Watson, T. (2018). The industrial internet of things (iiot): An analysis framework

Banerjee, A., Venkatasubramanian, K. K., Mukherjee, T., and Gupta, S. K. S. (2012). Ensuring safety, security, and sustainability of mission-critical cyber–physical systems. Proceedings of the IEEE, 100(1):283–299.

L. Ljung. Approaches to identification of nonlinear systems. In Control Conference (CCC), 2010 29th Chinese, pages 1–5, July 2010.

Müller, H., Litoiu, M., and Mylopoulos, J. (2016). Engineering cybersecurity in cyber physical systems. In Proceedings of the 26th Annual International Conference on Computer Science and Software Engineering, pages 316–320. IBM Corp

K. Angelopoulos, V. E. S. Souza, and J. Mylopoulos. Dealing with multiple failures in zanshin: a control-theoretic approach. In SEAMS 14, pages 165–174. ACM, 2014.

Shafi, Q. (2012). Cyber physical systems security: A brief survey. In 2012 12th International Conference on Computational Science and Its Applications, pages 146–150. IEEE

Morais, A., Hwang, I., Cavalli, A., and Martins, E. (2013). Generating attack scenarios for the system security validation. Networking science, 2(3-4):69–80.

Moore, A. P., Ellison, R. J., and Linger, R. C. (2001) Attack modeling for information security and survivability. Technical report,

Seid, E., Popov, O., and Blix, F. (2023). Security Attack Event Monitoring for Cyber Physical-Systems. In Mori, P., Lenzini, G., and Furnell, S., editors, Proceedings of the 9th International Conference on Information Systems Security and Privacy, ICISSP 2023, 2023, pages 722–732. SciTePress

Y. Brun, G. Marzo Serugendo, C. Gacek, H. Giese, H. Kienle, M. Litoiu, H. M¨uller, M. Pezz‘e, and M. Shaw. Software engineering for self-adaptive systems. chapter Engineering Self-Adaptive Systems Through Feedback Loops, pages 48–70. Springer-Verlag, Berlin, Heidelberg, 2009

A. Filieri, C. Ghezzi, A. Leva, and M. Maggio. Self-adaptive software meets control theory: A preliminary approach supporting reliability requirements. In 26th IEEE/ACM International Conference on Automated Software Engineering, ASE 2011, pages 283–292, 2011

A. Filieri, H. Hoffmann, and M. Maggio. Automated design of self-adaptive software with control-theoretical formal guarantees. In 36th International Conference on Software Engineering, ICSE '14, pages 299–310, 2014.

E. Camacho and C. Bordons. Model Predictive Control. Springer London, 2004.

V. Souza, A. Lapouchnian, and J. Mylopoulos. Requirements-driven qualitative adaptation. On the Move to Meaningful Internet Systems: OTM 2012, volume 7565 of Lecture Notes in Computer Science, pages 342–361. Springer Berlin Heidelberg, 2012.

S.Cheng, D. Garlan, and B. R. Schmerl. Architecture-based self-adaptation in the presence of multiple objectives. In Proceedings of the 2006 international workshop on Self-adaptation and self-managing systems, SEAMS 2006, pages 2–8, 2006.

V. E. S. Souza, A. Lapouchnian, W. N. Robinson, and J. Mylopoulos. Awareness requirements for adaptive systems. In 2011 ICSE Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS, pages 60–69, 2011.

A. Filieri, H. Hoffmann, and M. Maggio. Automated multi-objective control for self-adaptive software design. In Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering, ESECFSE 2015, pages 13–24, New York, NY, USA, 2015. ACM.

C. Klein, A. V. Papadopoulos, M. Dellkrantz, J. D¨urango, M. Maggio, K.-E. °Arz´en, F. Hern´andez-Rodriguez, and E. Elmroth. Improving cloud service resilience using brownout-aware load-balancing. In Reliable Distributed Systems (SRDS), 2014 IEEE 33rd International Symposium

P. Zoghi, M. Shtern, and M. Litoiu. Designing search based adaptive systems: a quantitative approach. In 9th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS 2014, Proceedings, pages 7–6, 2014.

H. Ghanbari, M. Litoiu, P. Pawluk, and C. Barna. Replica placement in cloud through simple stochastic model predictive control. In Cloud Computing (CLOUD), 2014 IEEE 7th International Conference on, pages 80–87, June 2014.

D. Kusic, J. Kephart, J. Hanson, N. Kandasamy, and G. Jiang. Power and performance management of virtualized computing environments via lookahead control. In Autonomic Computing, 2008. ICAC '08. International Conference on, pages 3–12, June 2008.

Q. Zhang, Q. Zhu, M. Zhani, and R. Boutaba. Dynamic service placement in geographically distributed clouds. In Distributed Computing Systems (ICDCS), 2012 IEEE 32nd International Conference on, pages 526–535, June 2012.

Gaggero and Luca Caviglione. 2016. Predictive control for energy-aware consolidation in cloud datacenters. IEEE Transactions on Control Systems Technology 24, 2 (Mar. 2016)

Hamoun Ghanbari, Marin Litoiu, Przemyslaw Pawluk, and Cornel Barna. 2014. Replica placement in cloud through simple stochastic model predictive control. In 2014 IEEE 7th International Conference on Cloud Computing

Dara Kusic, Jeffrey O. Kephart, James E. Hanson, Nagarajan Kandasamy, and Guofei Jiang. 2009. Power and performance management of virtualized computing environments via lookahead control. Cluster Comput. 12, 1 (2009)

Nilabja Roy, Abhishek Dubey, and Aniruddha Gokhale. 2011. Efficient autoscaling in the cloud using predictive models for workload forecasting. In 2011 IEEE International Conference on Cloud Computing (CLOUD'11). 500–507.

J. Maciejowski. Predictive Control: With Constraints. Prentice Hall, 2002.

K. Angelopoulos, V. E. S. Souza, and J. Mylopoulos. Capturing variability in adaptation spaces: A three-peaks approach. In Conceptual Modeling – ER 2015. Paul Johannesson and Mong Li Lee and Stephen W. Liddle and Oscar Pastor, 2015

J. Karlsson and K. Ryan. A cost-value approach for prioritizing requirements. Software, IEEE, 14(5):67–74, Sep 1997.

M.Gaggero and L. Caviglione. Predictive control for energy-aware consolidation in cloud datacenters. Control Systems Technology, IEEE Transactions on, pages 1–14, 2015.

Y. Wang and S. Boyd. Fast model predictive control using online optimization. Control Systems Technology, IEEE Transactions on, 18(2):267–278, March 2010.

Conway, J. (2016). The industrial internet of things: an evolution to a smart manufacturing enterprise. Schneider Electric.

Cailliau, A.van Lamsweerde, Runtime monitoring and resolution of probabilistic obstacles to system goals. In Software Engineering for Adaptive and Self-Managing Systems (SEAMS), 2017 IEEE/ACM

S. J. QinS. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. Control engineering practice, 11(7):733–764, 2003.