A multi-modelling based approach to assessing the security of smart buildings

***J.C. Mace***, *C. Morisset*†*, *K. Pierce*†*, *C. Gamble*†*, *C. Maple***, *J. Fitzgerald*†**

**University of Warwick, UK, john.mace@ncl.ac.uk,* †*Newcastle University, UK, charles.morisset@ncl.ac.uk*

Keywords: methodology, cyber-physical systems, INTO-CPS tool chain, co-simulation, adversary model.

Abstract

Smart buildings are controlled by multiple cyber-physical systems that provide critical services such as heating, ventilation, lighting and access control. These building systems are becoming increasingly vulnerable to both cyber and physical attacks. We introduce a multi-model methodology for assessing the security of these systems, which utilises INTO-CPS, a suite of modelling, simulation, and analysis tools for designing cyber-physical systems. Using a fan coil unit case study we show how its security can be systematically assessed when subjected to Man-in-the-Middle attacks on the data connections between system components. We suggest our methodology would enable building managers and security engineers to design attack countermeasures and refine their effectiveness.

1 Introduction

Modern buildings are often labelled as ‘smart’ due to the high level of automation incorporated into their critical systems providing heating, ventilation, and air-conditioning (HVAC), lighting, access control, and so forth. The use of network technologies to exchange data within a single system, between disparate systems, and with remote Internet-based systems means security is becoming a considerable problem for smart buildings [2]. For instance, many guests were reportedly locked inside their rooms after the electronic door locks were disabled in an Austrian hotel[1], the heating systems of several apartment blocks in Finland were disabled in the middle of winter after a DDoS attack[2], and up to 100 million credit card details were stolen when a US retailer’s IT network was accessed via the HVAC system[3]. Despite the rising number of reported attacks, security mechanisms within smart buildings can often be lacking. A 2016 survey conducted by the Electrical Contractors’ Association (ECA), the Chartered Institution of Building Services Engineers (CIBSE), and the Electrical Contractors Association of Scotland (SELECT) found almost 39% of building managers don’t take any steps to protect smart building systems from cyber threats [6].

Most smart building systems can be classified as cyber-physical systems (CPSs), comprised of software-based cyber controllers that interact with physical sensor and actuator devices. Furthermore, their inter-connection creates an Internet of Things, or IoT [23], meaning a complicated cyber-physical attack surface is presented: physical components can be accessed and tampered with thus impacting cyber elements, while digital controllers can be injected with fake data which can impact physical elements. Comprehending security requirements of CPSs is challenging especially where current security assessment processes are limited, coming with specific techniques that are applicable to very narrow fields and do not consider more holistic security considerations. In this context, it can be difficult to assess the impact of cyber-physical attacks and devise appropriate countermeasures [1].

One way to better understand the impact of security threats is to model both cyber and physical elements of a building system in a single multi-model. We therefore introduce a new multi-modelling methodology for analysing the security of smart buildings. Our methodology utilises INTO-CPS, an open tool chain for model-based design of CPSs [18]. Given an initial multi-model, the connections between constituent models that are vulnerable to attack are identified. We then introduce parameterised adversary models, which take in data being transmitted via these connections and potentially modifies it, akin to the Man-in-the-Middle (MITM) attacks on CPSs described in [16]. The parameters of an adversary model correspond to the different strategies an adversary can adopt. We then use the Design Space Exploration (DSE) facilities of INTO-CPS to automatically find the optimal adversary attack strategies. The simulation results can be used to design countermeasures against these strategies before refining their effectiveness using our methodology in an iterative way.

More explicitly, the contributions of this paper are as follows:

- A 6 step methodology to conduct security assessments of smart building CPSs subjected to MITM attacks.
- An illustration of our methodology using a fan coil unit

[1] http://www.dailymail.co.uk/news/article-4163886/Alpine-hotel-brings-locks-cyber-hacking.html

[2] https://boingboing.net/2016/12/02/ddos-attack-on-finnish-automat.html

[3] http://www.zdnet.com/article/anatomy-of-the-target-data-breach-missed-opportunities-and-lessons-learned/

(FCU) case study.

To the best of our knowledge, this is the first study to include systematic multi-modelling of CPSs together with security concepts, and suggest it provides the basis for more accomplished smart building security assessment processes.

The rest of the paper is structured as follows: Section 2 provides relevant background information on smart buildings and INTO-CPS, Section 3 outlines our multi-modelling methodology before illustrating its use by way of an FCU case study in Section 4. Related work in Section 5, and concluding remarks in Section 6.

2 Background

In this section we provide background information on smart buildings and general security risks to their building systems, before describing the INTO-CPS open tool chain for model-based design of cyber-physical systems.

2.1 Smart buildings

2.1.1 Building systems

The Internet of Things is allowing the development of a new class of smarter building that take advantage of automated processes to control various aspects of the facility including heating, ventilation, and air-conditioning (HVAC), lighting, security and access control, and safety and mobility, among other functions. The main motivations for this include improving operational efficiency, productivity, environmental sustainability, occupant health and safety, and reducing energy consumption. To realise these potential advantages, modern building systems integrate sensors to measure and collect data about the building environment (e.g. air temperature, humidity and occupancy). This data is transmitted to controllers which process it and, according to rules (the control logic), send control instructions to actuators to turn lights on or off, open or close air vents, and change a room's temperature, for example. This integration of software-based cyber controllers and physical sensors and actuators devices means many building systems can be classified as cyber-physical systems (CPSs).

Cyber and physical components typically, exchange data over some form of communications link, be it a link localised to a single system or a link between different systems. For example a building system may send data to controllers of disparate building systems, or to centralised building management workstations and servers using various wired and wireless communication media and protocols. Internet technologies enable connection to a building's common IT network for central processing, visualisation and data storage, as well as remote monitoring (e.g. outsourced facility management), and data retrieval, (e.g. weather forecasts). The use of open communication protocols, such as BACnet [3] and KNX [28], allow for data to be exchanged between building system components coming from different vendors [24].

2.1.2 Security issues

Building systems present a complex cyber-physical attack surface: physical components can be accessed and tampered with, while digital controllers can be injected with fake data. Whilst understanding both the potential cyber and physical vulnerabilities is difficult, it is not as difficult as attempting to assess the impact of a successful attack. For instance, a cyber-physical attack could send destructive commands over the control network to equipment that changes its configuration above dangerous levels for which the equipment has not been designed (e.g. too high a pressure or temperature). Understanding such an attack requires understanding the impact on both the control network (cyber) and the equipment (physical), and crucially, their interconnection.

Building systems typically represent 40% of a building's energy usage. If lighting is included, this number approaches 70%. It is therefore critical to manage energy demand correctly. However, building systems configured or operated incorrectly are believed to account for a 20% increase in building energy usage over correctly configured systems [27]. Cyber-physical attacks on 'correct' systems could therefore increase energy consumption and incur extra financial costs as well as costs to operations and reputation (e.g. increased maintenance, reduced health and safety, and lower environmental sustainability). It is therefore crucial to systematically assess the impact of different cyber-physical attacks on building systems by adversaries coming with different capabilities.

2.2 INTO-CPS

INTO-CPS is an Integrated Tool chain for model-based design of CPSs [18]. Here we outline how multi-models of CPSs are constructed, how co-simulations are performed using those models, and how a range of simulations can be performed using Design Space Exploration.

2.2.1 Multi-modelling

To model a CPS it is necessary to capture the behaviour of both the cyber and physical elements, however, the nature of these behaviours mean that they are best modelled using different modelling notations and tools. In general, the cyber elements are modelled using a Discrete Event (DE) notation since this provides appropriate abstractions for describing the data structures and control flows that may be found in a cyber controller. At the same time, the physical elements are better described in a Continuous Time (CT) environment where behaviour is captured in the form of differential equations along with suitable integration methods [20]. INTO-CPS makes use of the Overture tool[4] and the VDM-RT language [21] for DE modelling, while 20-Sim[5] and OpenModelica[6] provide CT modelling environments.

[4] http://overturetool.org/

[5] http://www.20sim.com/

[6] https://openmodelica.org/

Modelling of a CPS starts with it being decomposed into cyber and physical elements, where those elements present interfaces allowing them to input data from and output data to each other. Each of these elements would then be modelled by domain experts, making use of the appropriate DE or CT environment. When the individual modelling activities are complete, it is then possible to recombine the elements to form a holistic model of the CPS. To overcome the differences between the modelling tools, INTO-CPS expects that the individual models are exported from their modelling tool using the Functional Mock-up Interface (FMI) standard. A model exported according to this standard is called a Functional Mockup Unit (FMU). An INTO-CPS multi-model of a CPS then is the combination of a set of FMUs along with multi-model configuration file detailing how to connect their inputs and outputs.

2.2.2 Co-simulation

When the multi-model is constructed it is then possible to perform a co-simulation of the multi-model to observe the behaviour of the CPS. INTO-CPS includes a Co-simulation Orchestration Engine (COE) that is responsible for reading the multi-model configuration, launching the models enclosed in the FMUs, exchanging data between the FMUs and, importantly, determining the appropriate time steps each model should take to keep them all synchronised and stable. The user interacts with the multi-model configuration and the COE via the INTO-CPS application and it is in this application where the user may define the length of the simulation, the time stepping method, and actually launch a co-simulation. The INTO-CPS application also provides a live plotting facility, where the user may choose to plot variables of the multi-model during simulation so as to observe the modelled behaviour.

2.2.3 Design Space Exploration

It is likely that there are many choices to be made when designing a CPS and these choices will affect the resulting performance of the CPS. Choices could include physical properties of the CPS, such as the thickness of walls in a building or the number or placement of sensors within a room, or they could regard cyber properties such as the choice of algorithm controlling heating or the frequency at which sensors are sampled. These choices along with the options for each define the design space for the CPS. One use for a multi-model then is to allow the engineer to explore the design space to find design options that are optimised with respect to one or more performance measures. Many of the design choices can be left open by the domain experts that produced the original models by exposing them as parameters of the resulting FMUs, in which case the user has the option to make use of the DSE facilities included in INTO-CPS to automatically explore the design space [11].

As a minimum, a DSE requires the definition of three aspects. The first aspect, parameters, is where we describe which parameters the DSE search may change and also gives a list of values each parameter may take. These parameters define the design space that is to be searched.

The second and third aspects relate to how we measure performance of a system and how we compare different designs using those measures. INTO-CPS simulations produce results in three forms, live graph plots of variables during a simulation, logs of monitored variables in CSV format and also 3D visualisations of the models if the user has created one. Since DSE is likely to run a great many simulations it is not practical for a user to observe all the live plots or 3D visualisations and so DSE makes use of Objective scripts that process the CSV simulation logs to produce objective values that characterise the performance of a CPS during simulation. Such objective functions might compute the total energy consumed by a system or the maximum deviation of a variable from an acceptable value. Once the objective values are computed for each design, they may then be used to compare different designs. If there is only a single performance measure than results may simply be placed in an list, ordered by that measure, to find the best, however, if there are multiple measures then a different means for comparison must be used. In the latter case, INTO-CPS makes uses the Pareto method to present the user with a non-dominated set of best designs [9].

3 Multi-modelling methodology

In this section we introduce our multi-modelling methodology by first giving an overview of its six steps before describing each of those steps in more detail. We present this methodology from the viewpoint of a security engineer, and we therefore assume that a multi-model for the CPS already exists (see above), albeit with no security specific elements. In a nutshell, the six steps are:

1. Identify data connections between constituent models.
2. Create an adversary model for each type of connection.
3. Identify relevant security metrics.
4. Design model for missing metrics.
5. Run DSE to identify optimal attacks.
6. Propose and design counter measures.

3.1 Identify data connections

The connections in a multi-model represent exchanges of data between constituent models, but not all of those exchanges represent connections that are vulnerable to MITM attacks. The first step in the methodology is therefore to establish which connections are potentially vulnerable. This identification relies on the nature of the connection and also any implementation details. For instance if the connection represents communications between a sensor and a controller then this connection might be vulnerable depending on whether encryption is used and if an attacker could have a physical access. However, a connection representing energy transfer between a wall model and the outside environment model could not be attacked in this way.

3.2 Create adversary model

Given a data connection, the next step is to define an adversary model for that connection, i.e., to make explicit how an attacker can change the data exchanged over the connection. As suggested in Section 1, many physical and cyber attacks may be possible on a CPS, possibly coming with different parameters and interfaces. In the most general case, the attacker could potentially change a value on a connection in any way they like. However, in practice, we might want to encode the actual supposed qualities of the attacker. For instance, an attacker might only be able to send fake data, without blocking the real data, or an attacker might only be able to change environmental data by physically changing the environment (e.g. putting a heat source by a temperature sensor).

3.3 Identify relevant metrics

In general, security metrics might be different from the functional metrics used for the design of the multi-model. The selection of metrics for the DSE step (see below) therefore depends on the characterisation of the expected impact of attacks on the system: financial cost by increasing energy consumption, denial of service, loss of information, etc. Irrelevant metrics can be removed, since DSE can be computationally expensive, while missing metrics need to be designed.

3.4 Design for missing metrics

It is possible that the original multi-model was not designed to output metrics that are required by the security analysis, in which case it will be necessary to either modify one of the existing constituent models or add a new model. For example, suppose we want to measure the wear on a CPS component, since the attack we wish to simulate attempts to maximise the components usage. The original model does not consider component usage, so this metric is therefore added.

3.5 Run Design Space Exploration

With the model modified to support the security analysis, we can now run a DSE to elicit the 'optimal' attacker, where optimal might be determined using antagonistic metrics. For example the adversary might want to maximise a specific security metric (e.g. maximum usage), while minimising likelihood of the attack being detected. The DSE generates a set of ranked Pareto curves where rank 1 is the set of 'best' adversary models in respect to simulation objectives and the lowest rank is the set of 'worst' adversaries.

3.6 Propose countermeasures

Having analysed results of a DSE, it may be desirable to implement countermeasures to remove, or at the very least, alleviate the impact of a MITM attack. Countermeasures may involve simply tweaking existing CPS operational settings; running a DSE to find the optimum from a range of possible operational settings is one method. Alternatively, it may be necessary to design new countermeasures and implement them within one or more constituent models of the CPS. Introducing countermeasures can mitigate an attack but also negatively impact the operation of a CPS. By employing our methodology in an iterative way, a countermeasure, or a set of countermeasures, can be continually refined and tested to find a satisfactory compromise between the security of a CPS and its operational effectiveness.

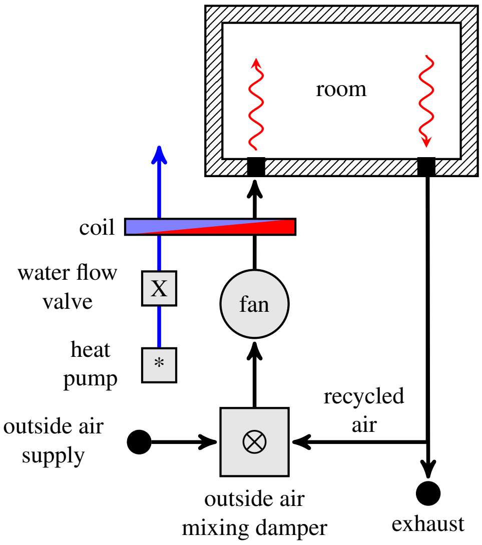

Fig. 1: Overview of fan coil unit (FCU) example.

4 Fan coil unit case study

In this section we illustrate our methodology with a fan coil unit (FCU) case study. We describe an FCU, a multi-model of the FCU created using INTO-CPS, and show a co-simulation of the FCU's normal operation. We then consider a MITM attack which aims to expedite wear on the FCU, and use our methodology to assess the security of the FCU under this attack.

4.1 Fan coil unit

Figure 1 provides an abstracted overview of an FCU example taken from [10]. The FCU contains a *fan* that draws air into the unit then blows it over a cooling/heating *coil*. The air comes out of the FCU, either heated or cooled depending on the coil temperature, and enters a *room*. Water flows into the coil from a *heat pump* which is reversible, and can work in either direction to either heat or cool the water supply. A *water flow value* controls the rate of water flow from the heat pump to the coil, thus controlling the temperature change rate of the coil. A cyber *controller* is able to alter both the fan speed and the water flow value. Each FCU is provided with a small *outside air supply* to ensure adequate ventilation, which is mixed in the *outside air mixing damper* with air recirculated from the room. Any excess recirculated air leaves the system via the *exhaust*.

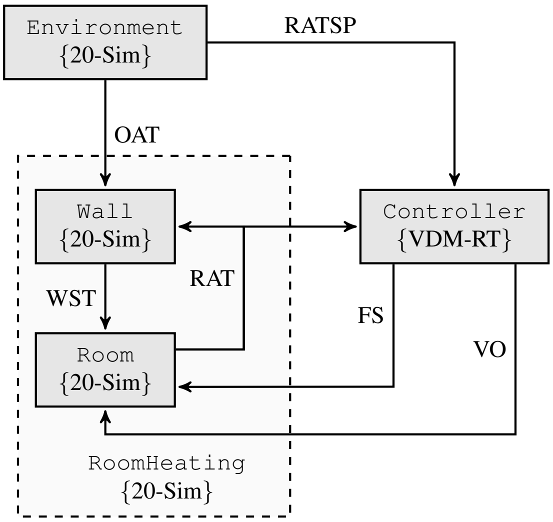

Fig. 2: Multi-model of the fan coil unit (FCU) example.

In addition, the room temperature is affected by the outside air temperature, the thermal conductivity of the room's walls and windows, and the room air temperature set-point. We assume the latter is set to 21°c by default during any working day which can be altered by room occupants by +/- 3°C in 0.5°c increments. These elements together constitute the environment of the FCU.

4.1.1 Fan coil unit multi-model

Figure 2 shows a multi-model of an FCU with 3 constituent models: `Environment`, `RoomHeating`, and `Controller`. The `Environment` and `RoomHeating` are continuous time models, created in 20-Sim, representing physical elements of the FCU . The `RoomHeating` model is comprised of two sub-models: `Wall` and `Room`. The `Controller` represents the cyber element of the FCU and is a discrete event model encoded in VDM-RT. All three constituent models are exported as FMUs which are connected using the INTO-CPS application to create an FCU multi-model.

The `Environment` model encapsulates the room air temperature set-point (RATSP) and the outside air temperature (OAT). The RATSP and OAT are communicated to the `Controller` and `Wall` models respectively. The `Controller` model inputs the RATSP and the current room air temperature (RAT) and outputs settings for the fan and water flow value labelled as fan speed (FS) and valve open (VO). The `Wall` model inputs the OAT and RAT and outputs the wall surface temperature (WST) to the `Room` model. The `Room` model inputs the WST, FS, and VO and outputs the RAT.

4.1.2 Fan coil unit co-simulation

Figure 3 shows a slice of the results of running a simulation of the FCU. The RAT (dark red line) starts at 15°C, and begins to rise around time = 510 when the RATSP (orange line) is 21°C and the controller increases FS (green line). When RAT reaches RATSP, FS reduces and then varies to maintain the desired air temperature. The wear on the FCU, represented by the fan usage (light red line), is also shown. Note that fan usage increases more rapidly at higher fan speeds.

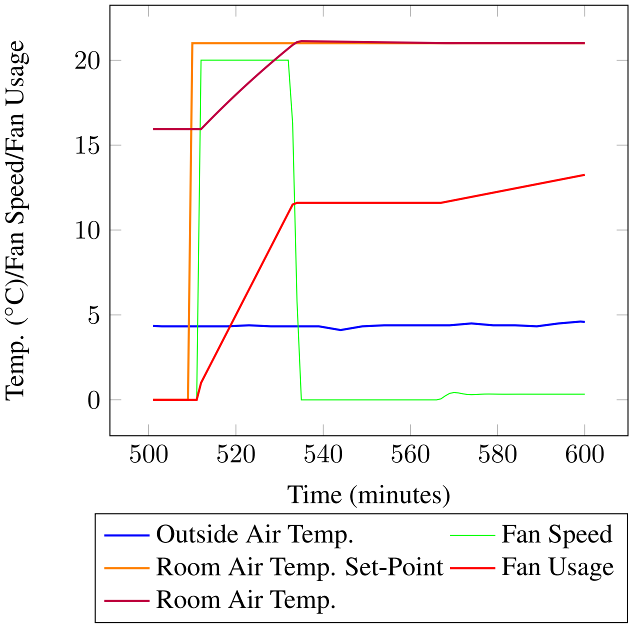

Fig. 3: Co-simulation snippet between 500 and 600 minutes of FCU's normal operation.

4.2 Fan coil unit security analysis

We now illustrate how our methodology is used to assess the security of the FCU being subjected to a MITM attack aiming to maximise wear on the FCU, driving up energy and maintenance costs as a result.

4.2.1 Identify data connections

Given the multi-model illustrated in Figure 2, the first step is to identify connections that are potentially vulnerable to MITM attacks. Four such connections are identified in this case: `Environment` → `Controller` transmitting RATSP, `Room` → `Controller` transmitting RAT, and two `Controller` → `Room` connections transmitting FS and VO respectively. For simplicity, we assume these data connections are unencrypted. All remaining connections involve physical data (i.e. energy) transfer and are not susceptible to a MITM attack.

4.2.2 Create adversary model

For illustration purposes we consider a MITM attack on a single data connection; a more complex attack is considered in Section 4.3. Gaining control of the connection between the `Environment` and `Controller` enables an adversary to intercept, and potentially modify, the RATSP being transmitted. We assume the adversary's objective is to expedite the

Algorithm 1 Adversary model function $attack$ executed at frequency f_a.

```
RATSP_in ← 0
RATSP* ← 0
function ATTACK($\perp_t$,$\top_t$)
    RATSP_in = $input$(RATSP)
    if RATSP_in $> 0$ then
        if RATSP* $<$ RATSP_in then
            RATSP* ← RATSP_in + $\top_t$
        else
            RATSP* ← RATSP_in - $\perp_t$
        end if
    else
        RATSP* ← 0
    end if
    $output$(RATSP*)
end function
```

wear on the FCU by maximising fan usage, thus increasing costs from energy requirements, maintenance demands, etc.

By continually increasing and decreasing the RATSP at a given frequency, the adversary can force the FCU's fan to oscillate, that is, switch on and off. We therefore consider a parameterised DE adversary model $\mathcal{M}$ which can launch attack strategies of the form $(f_a, \top_t, \perp_t)$, where f_a is the frequency at which RATSP should be modified, $\top_t$ is the temperature increase to RATSP, and $\perp_t$ the temperature decrease. Algorithm 1 provides the $attack$ function encoded in $\mathcal{M}$, which is executed at frequency f_a. Note, RATSP is only modified if RATSP > 0.

To connect adversary model $\mathcal{M}$ to the FCU multi-model, its interface is implemented allowing it to input and output the RATSP data type. Figure 4[7] shows how $\mathcal{M}$ is placed between the `Environment` and `Controller` of the original FCU multi-model to prime the MITM attack. The interfaces of these two constituent models are not modified. The connection between $\mathcal{M}$ and `Controller` is labelled RATSP*, whose value may or may not be equal to RATSP.

4.2.3 Identify relevant metrics

As the MITM attack we are modelling with $\mathcal{M}$ involves modifying RATSP, which in turn impacts FS, clearly both of these metrics are relevant and remain in the model. The metrics OAT and WST are also necessary as they influence the RAT which, together with the RATSP, impacts FS and VO set by `Controller`. An energy metric E has been implemented by domain experts in `Controller`. We consider a more generic metric for wear on the FCU in terms of fan usage in Section 4.2.4 which comes with minimal computational overhead and therefore remove E from the multi-model.

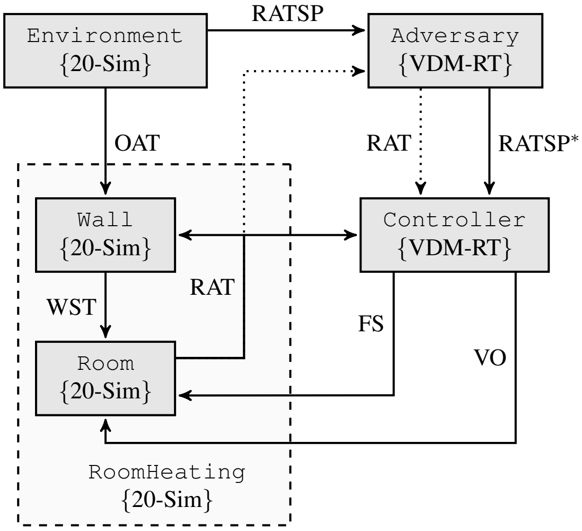

Fig. 4: Multi-model of FCU with `Adversary` $\mathcal{M}$ intercepting RATSP, and $\mathcal{M}^*$ intercepting both RATSP and RAT.

4.2.4 Design for missing metrics

We add a metric FU to `Controller` denoting the running total of FCU fan usage during a given period of operation. For illustration purposes it is enough to consider this simple metric to indicate the wear on the FCU when subjected to different MITM attacks. It can also indicate how much the fan's usage can be reduced without impacting operational requirements when introducing attack countermeasures. The FU metric encapsulates many different concepts including wear, energy use, maintenance demands, etc. Each concept could be represented by separate metrics but with increased computational overheads.

The FU metric is designed as follows: if the fan is switched on, that is if FS $= 0$ at simulation time step i and FS > 0 at simulation time step $i+1$, then a 'switch on' penalty of +0.5 is incurred. The following 'usage costs' are also incurred at each simulation time step i depending on the FS at time step i.

fan speed (FS)	usage cost
$0 <$ FS ≤ 5	+0.05
$5 <$ FS ≤ 10	+0.10
$10 <$ FS ≤ 15	+0.30
$15 <$ FS	+0.50

We have selected arbitrary usage costs which increase as FS increases. Note, the maximum value for FS is 20.

4.2.5 Run design space exploration

In the DSE configuration file, two objectives are stated for each simulation: compute the total fan usage, and compute the average temperature deviation of the RAT from the RATSP. We write μ_f and σ_t respectively to denote the values these two objectives. The reason for computing μ_f is straightforward as the

[7] The dotted connections can be disregarded at this stage.

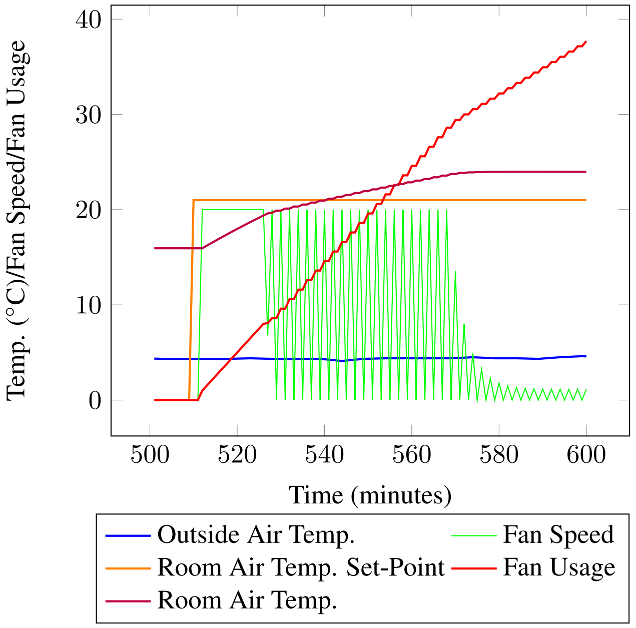

Fig. 5: Co-simulation snippet of MITM attack strategy α_1 using adversary model $\mathcal{M}$, where $f_a = 1$, $\top_t = 3$ and $\bot_t = 1.5$.

aim of the adversary is to maximise fan usage. Average temperature deviation is also chosen as we assume an adversary wants to avoid detection, that is by minimising the violation to user comfort over the simulation period.

We write X_j for the result set of all steps in the simulation of attack strategy α_j. With INTO-CPS, the result set X_j is outputted as a `.csv` file. For each simulation time step i, the result $x_i \in X_j$ is a tuple indicating the values of FCU metrics at time step i, denoted as (FU$_i$,RAT$_i$,RATSP$_i$,OAT$_i$). Trivially, as FU never decreases, total fan usage is equivalent to FU of the final simulation time step, that is $\mu_f = \text{FU}_{|X_j|}$. Total fan usage is computed automatically by INTO-CPS by using the `objectiveType 'Max'` function together with the `.csv` file column ID 'FU' within the DSE configuration file.

To compute the average temperature deviation of attack strategy α_j, we first discount any $x_i \in X_j$ where RATSP$_i = 0$. This gives result set $R_j \subseteq X_j$, where $R_j = \{x_i \in X_j \mid \text{RATSP}_i \neq 0\}$. The average temperature deviation σ_t for α_j is then given by:

$$\sqrt{\frac{\sum_{i=1}^{|R_j|}(\text{RAT}_i - \text{RATSP}_i)^2}{|R_j|}}$$

The average temperature deviation is computed automatically by INTO-CPS by encoding the above calculation into an imported user metric script.

For each attack strategy simulation, the start and end times are set to the equivalent of 500 minutes and 1050 minutes in actual time. This approximates to one working day, that is, 8.30am to 5.30pm. A much longer simulation period could be implemented, say over a week or month, but it is enough to illustrate our approach to simulate the FCU operating over a single working day. Simulation time step size is set to 1 minute such that 550 results are generated per simulation, that is $|X_s| = 550$.

Figure 5 shows a simulation snippet between 500 and 600 minutes of a single attack strategy α_1, where $f_a = 1$ (minute), $\top_t = 3$ (°C), and $\bot_t = 1.5$ (°C). Comparing this plot to Figure 3 where the FCU is operating normally, it is clear to see how the attack strategy increases the FCU usage by making the fan speed oscillate. After 600 minutes of normal operation, FU $\approx$ 13, whereas when being attacked, FU $\approx$ 38, more than double in the same period.

In the DSE configuration file, the parameter ranges for adversary model $\mathcal{M}$ are as follows:

- f_a : [0.25,0.50,0.75,1.00,1.25,1.50,1.75,2.00]
- $\top_t$: [0.0,0.5,1.0,1.5,2.0,2.5,3.0]
- $\bot_t$: [0.0,0.5,1.0,1.5,2.0,2.5,3.0]

Taking all parameter combinations into account, the DSE comprises of running 392 simulations and outputting the subsequent result sets.

The DSE simulations are ranked to generate a Pareto front whose points maximise fan usage (objective 1) and minimise temperature deviation (objective 2). The Pareto front produced by the DSE is shown in Figure 6 (red line) and whose endpoints are given in Table 1. The Pareto front end points are essentially the two optimal attack strategies denoted as α_1 and α_2. Attack strategy α_1 satisfies objective 1 by maximising the total fan usage ($\mu_{fan} = 153.20$), but not objective 2; the average temperature deviation is also maximised ($\sigma_{temp} = 2.83$). Alternatively, attack strategy α_2 satisfies objective 2 by minimising the average temperature deviation ($\sigma_{temp} = 0.68$), but not objective 1; the total fan usage is also minimised ($\mu_{fan} = 23.50$).

The Pareto front effectively offers the optimal trade-offs between maximising fan usage and minimising average temperature deviation. An adversary wanting to incur maximum FCU usage in the shortest time, and with little concern for detection, would arguably use an attack strategy on the Pareto front close to α_1. On the other hand, an adversary willing to 'play the long game' and incur maximum FCU usage with a low risk of detection, would arguably use an attack strategy on the Pareto front close to α_2.

Interestingly, as seen in Figure 6 the attack can generate a total fan usage $\mu_f \approx 100$ while the average temperature deviation σ_t is kept below 1. The optimal attack strategy for this, $\alpha = (0.75, 0.5, 0.5)$ results in $\mu_f = 101.85$ and $\sigma_t = 0.813$. In order to increase μ_f any attack strategy will increase σ_t, such that $2 \lessapprox \sigma_t \lessapprox 3$.

A special case must be noted, that is, for this particular MITM attack to have any impact on FCU fan usage, RAT must be less than RATSP when RATSP > 0. Clearly, if RAT $>$ RATSP > 0 then it is unnecessary for the FCU fan to switch on in order to

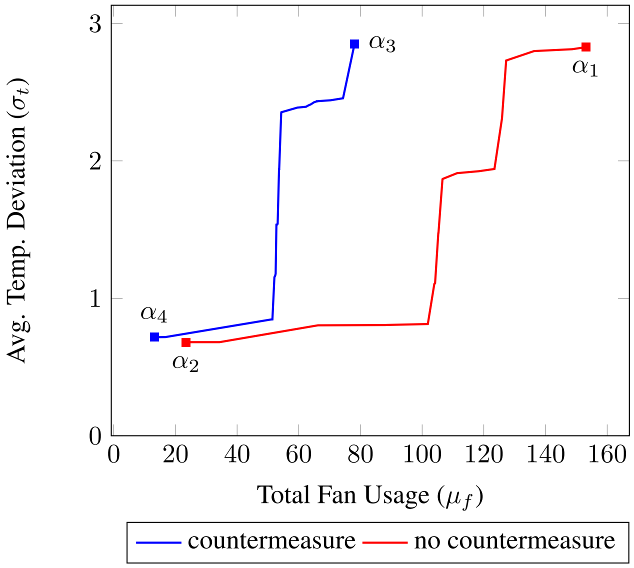

Fig. 6: DSE Pareto fronts illustrating adversary $\mathcal{M}$ optimal attacks with and without countermeasure.

raise RAT towards the RATSP. If this condition holds for the duration of the attack, the attack can be considered a failure as no fan usage will have been incurred. A more complex MITM attack could force the FCU's fan to oscillate continually, even when fan operation is not required.

4.2.6 Propose countermeasures

Having observed the DSE results presented in Section 4.2.5 it may be desirable to implement countermeasures to alleviate the impact of the MITM attack. For illustration purposes we consider decreasing the frequency, in which `Controller` polls the incoming data connection for the current RATSP. More precisely, the countermeasure reduces `Controller` polling frequency from 1 to 2 minutes.

Figure 7 shows a co-simulation snippet between 500 and 660 minutes of attack strategy $\alpha_1 = (1, 3, 1.5)$ on the FCU with the implemented `Controller` frequency countermeasure. Comparing this plot to Figure 5 where no countermeasure is implemented, it is clear to see how the countermeasure has decreased fan usage. The Pareto front produced by the DSE is shown in Figure 6 (blue line), whose endpoints are given in Table 1, denoted as attack strategies α_3 and α_4.

Positively, the countermeasure reduces the maximal total fan usage μ_f caused by the attack (using α_3) by almost half, and fractionally increases the minimal average temperature deviation σ_t (using α_4), thus increasing the chance of attack detection. Negatively, the countermeasure in some cases can prevent the FCU from raising the RAT to the required RATSP as seen in Figure 7; RAT only reaches a maximum of 20°C before beginning to fall. This would imply a trade-off must be sought between FCU operation and security.

	α	μ_f	σ_t	f_a	$\top_t$	$\perp_t$
Attack	α_1	153.20	2.83	1.00	3.00	1.50
	α_2	23.50	0.68	2.00	0.00	0.50
Countered	α_3	78.05	2.85	2.00	3.00	3.00
	α_4	13.20	0.72	1.50	0.00	0.50

Table 1: Pareto front endpoints from DSE indicating optimal attacks and optimal countered attacks on fan coil unit.

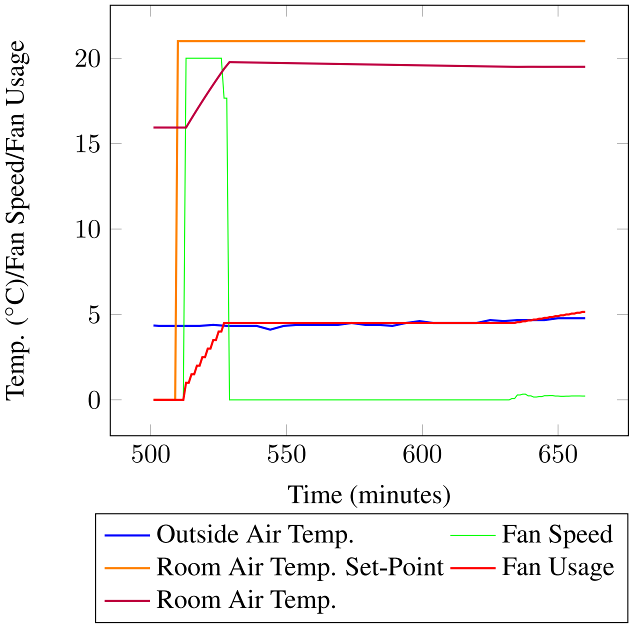

Fig. 7: Co-simulation snippet of MITM attack strategy α_1 using adversary model $\mathcal{M}$, where $f_a = 1$, $\top_t = 3$ and $\perp_t = 1.5$, and FCU `Controller` frequency countermeasure.

4.3 More complex models

So far we have consider a MITM attack on a single data connection. We now give a glimpse of how our methodology can be used to analyse more sophisticated adversary models. Figure 4 shows how adversary $\mathcal{M}$ is extended to intercept both RATSP and RAT (dotted arrow) to form adversary $\mathcal{M}^*$. With this attack scenario, $\mathcal{M}^*$ monitors RAT to decide when to modify the RATSP; in this case whenever RAT $<$ RATSP $+ 1$. It is assumed the risk of detection is too great for the adversary when the RAT moves above this point. Figure 8 shows a co-simulation snippet between 500 and 660 minutes of attack strategy $\alpha_1 = (1, 3, 1.5)$ on the FCU under adversary $\mathcal{M}^*$. Note how the attack causes the fan speed to oscillate only when RAT is below 22°C.

5 Related Work

Many model-based tools focus on single formalisms [4], which can be a barrier to modelling heterogeneous systems where diverse models from multiple sources would be more appro-

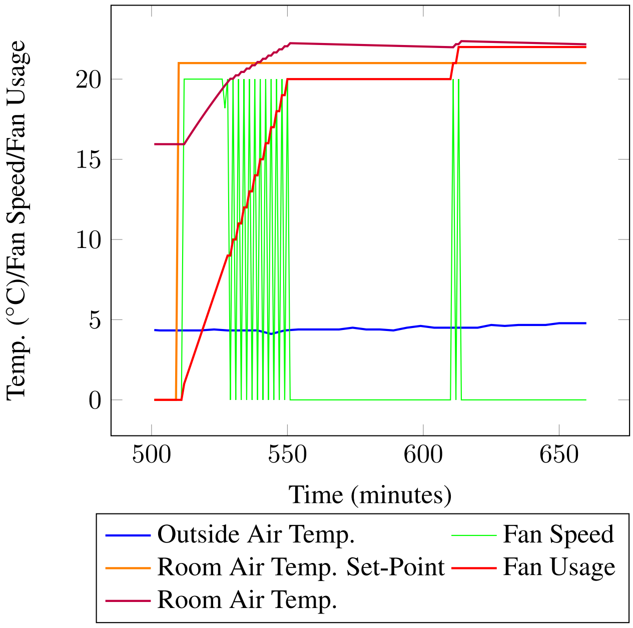

Fig. 8: Co-simulation snippet of MITM attack strategy α_1 using adversary model $\mathcal{M}^*$, where $f_a = 1$, $\top_t = 3$ and $\bot_t = 1.5$.

priate [5]. Connecting and analysing models through co-simulation is a promising alternative to single-formalism approaches, with a large variety of approaches currently being researched [13]. Bespoke solutions for co-simulation of infrastructure models have been demonstrated [22, 12, 17], including those using network simulators to mimic real communications [14]. The INTO-CPS approach is to provide a general co-simulation framework based on an open protocol, which seeks to avoid vendor lock-in and provides protection for intellectual property by supporting ‘black box’ models [19]. The INTO-CPS approach has been successfully applied in a number of industry settings, by companies outside of the original project, in areas including marine propulsion [26], autonomous planetary rovers [7], and manufacturing [25].

Given the aforementioned challenges in understanding the vulnerabilities and impact of breaches in complex CPSs such as those in modern buildings, it is important to use appropriate techniques to assist in security assessments. Examples of methodologies for security assessment in industrial control system settings include [15]. This approach is based upon an introductory presentation on security assessments (an integral part of the methodology since the methodology is designed for non-experts as well as experts); a security assessment form; and mind maps used to provide a visual presentation of any issues identified. It is argued that this approach gives much broader results (such as recommendations for audit logging) than a penetration test, and, in contrast to threat modelling, requires the assessor to include possible mitigation strategies. However, without the involvement of security professionals, the effectiveness of the results could be limited. The approach to physical security in the methodology is restricted to unauthorised access, rather than a more holistic security consideration.

In contrast to the work in [15], the work presented in [29] (which features two authors in common with the former paper) is designed to be used by security professionals. The authors present a threat modelling methodology that aims to address the issue that many techniques are specific and only applicable in very narrow fields, or generic and fail to generate data that is useful for automated processing.

More recent work has seen the development of a hybrid, distributed simulation platform for cyber-security analysis of large-scale critical infrastructure systems, [8]. The authors integrate simulated, emulated and real systems upon which vulnerability analysis could be performed.

6 Conclusion

This paper presents the first multi-modelling methodology for capturing and analysing Man-in-the-Middle attacks against cyber-physical systems (CPSs), illustrated on a fan coil unit in a smart building. Our approach particularly enables the design of an optimal attacker, and the modelling and assessment of appropriate countermeasures.

We pave the way for an exciting range of CPS security modelling, and several strands of future works are possible. Firstly, we can increase the complexity of the multi-model (e.g. attack on heating system of a data centre, for which the controller is arguably more complex), as well as the complexity of the adversary (e.g. attacks distributed over several components with causal failure, modelling of real world communication protocols using INTO-CPS Ether, possibly including encryption). The future development of the INTO-CPS tool chain will also improve the depth of the security assessment. In addition to the INTO-CPS Ether, mentioned above, the notion of scenario sweeping in INTO-CPS DSE will enable to change aspects of the simulation, such as the outside air temperature profile or the user temperature set-point profile, such that the attack and mitigation may be examined under a variety of conditions rather than just one. Finally, we plan to apply our approach on existing and deployed CPS, such as those provided with Newcastle University’s Urban Sciences Building and the many building systems within it to explore and assess.

Acknowledgements. The research in this paper was supported by UK EPSRC through ‘Newcastle Urban Sciences Building IoT’, a PRF project funded under grants EP/N02298X/1 and EP/N02334X/1 within the PETRAS IoT Research Hub.

References

[1] Y. Ashibani and Q. H. Mahmoud. Cyber physical systems security: Analysis, challenges and solutions. *Computers & Security*, 68:81 – 97, 2017.

[2] H. Boyes. Security, privacy, and the built environment. *IT Professional*, 17(3):25–31, 2015.

[3] S. T. Bushby and H. M. Newman. BACnet Today: Significant new features and future enhancements. *ASHRAE Journal*, 44(10):10–17, 2002.

[4] L. P. Carloni, R. Passerone, A. Pinto, and A. L. Sangiovanni-Vincentelli. Languages and tools for hybrid systems design. *Foundations and Trends in Electronic Design Automation*, 1(1/2), 2006.

[5] P. Derler, E. A. Lee, and A. Sangiovanni-Vincentelli. Modeling Cyber-Physical Systems. *Proc. of the IEEE (special issue on CPS)*, 100(1):13 – 28, 2012.

[6] ECA, CIBSE & SELECT. Connected technology survey for clients, 2016.

[7] S. Feo-Arenis, M. Verhoef, and P. G. Larsen. The marsrover case study modelled using into-cps. In *Proc. of the 15th Overture Workshop: New Capabilities and Applications for Model-based Systems Engineering*, pages 130–144, 2017.

[8] M. Ficco, M. Chora, and R. Kozik. Simulation platform for cyber-security and vulnerability analysis of critical infrastructures. *Journal of Computational Science*, 22:179 – 186, 2017.

[9] J. Fitzgerald, C. Gamble, R. Payne, and B. Lam. Exploring the cyber-physical design space. In *INCOSE Int. Symp.*, volume 27, pages 371–385, 2017.

[10] J. Fitzgerald, C. Gamble, R. Payne, P. G. Larsen, S. Basagiannis, and A. E.-D. Mady. Collaborative model-based systems engineering for cyber-physical systems, with a building automation case study. *INCOSE Int. Symp.*, 26(1):817–832, 2016.

[11] C. Gamble. Comprehensive DSE Support. Technical report, INTO-CPS Deliverable, D5.3e, December 2017.

[12] M. Garau, G. Celli, E. Ghiani, F. Pilo, and S. Corti. Evaluation of smart grid communication technologies with a co-simulation platform. *IEEE Wireless Communications*, 24(2):42–49, 2017.

[13] C. Gomes, C. Thule, D. Broman, P. G. Larsen, and H. Vangheluwe. Co-simulation: State of the art. Technical report, feb 2017.

[14] Z. Hill, S. Chen, D. Wall, M. Papa, J. Hale, and P. Hawrylak. Simulation and analysis framework for cyber-physical systems. In *Proc. of the 12th Conf. on Cyber and Inf. Security Research*, CISRC, pages 7:1–7:4, 2017.

[15] A. Hristova, R. Schlegel, and S. Obermeier. Security assessment methodology for industrial control system products. In *Proc. of the 4th Annual IEEE Int. Conf. on Cyber Technology in Automation, Control and Intelligent Systems*, CYBER, pages 264–269, 2014.

[16] I. Jovanov and M. Pajic. Relaxing Integrity Requirements for Attack-Resilient Cyber-Physical Systems. *ArXiv e-prints*, July 2017.

[17] H. Kim, K. Kim, S. Park, H. Kim, and H. Kim. Cosimulating communication networks and electrical system for performance evaluation in smart grid. *Applied Sciences*, 8(1), 2018.

[18] P. G. Larsen et al. Integrated tool chain for model-based design of cyber-physical systems: The INTO-CPS project. In *Proc. of the 2nd Int. Workshop on Modelling, Analysis, and Control of Complex CPS*, pages 1–6, 2016.

[19] P. G. Larsen, J. Fitzgerald, J. Woodcock, C. Gamble, R. Payne, and K. Pierce. Features of integrated model-based co-modelling and co-simulation technology. In *Proceedings of the 1st Workshop on Formal Co-Simulation of Cyber-Physical Systems*, September 2017.

[20] P. G. Larsen, J. Fitzgerald, J. Woodcock, R. Nilsson, C. Gamble, and S. Foster. Towards semantically integrated models and tools for cyber-physical systems design. In *Int. Symp. on Leveraging Applications of Formal Methods*, pages 171–186, 2016.

[21] P. G. Larsen, K. Lausdahl, N. Battle, J. Fitzgerald, S. Wolff, S. Sahara, M. Verhoef, P. W. V. Tran-Jørgensen, and T. Oda. The VDM-10 Language Manual. Technical Report TR-2010-06, The Overture Open Source Initiative, April 2010.

[22] H. Lin, S. Sambamoorthy, S. Shukla, J. Thorp, and L. Mili. Power system and communication network co-simulation for smart grid applications. In *IEEE PES Innovative Smart Grid Technologies*, pages 1–6, 2011.

[23] C. Maple. Security and privacy in the internet of things. *Journal of Cyber Policy*, 2(2):155–184, 2017.

[24] H. Merz, T. Hansemann, and C. Hbner. *Building Automation: Communication systems with EIB/KNX, LON and BACnet*. Springer, 2009.

[25] M. Neghina, C.-B. Zamrescu, P. G. Larsen, K. Lausdahl, and K. Pierce. A Discrete Event-First Approach to Collaborative Modelling of Cyber-Physical Systems. In *Proc. of the 15th Overture Workshop: New Capabilities and Applications for Model-based Systems Engineering*, pages 116–129, September 2017.

[26] N. Pedersen, K. Lausdahl, E. V. Sanchez, P. G. Larsen, and J. Madsen. Distributed Co-Simulation of Embedded Control Software with Exhaust Gas Recirculation Water Handling System using INTO-CPS. In *Proc. of the 7th Int. Conf. on Simulation and Modeling Methodologies, Technologies and Applications*, pages 73–82, July 2017.

[27] K. W. Roth, D. Westphalen, J. Dieckmann, S. D. Hamilton, and W. Goetzler. Energy consumption characteristics of commercial building hvac systems volume iii: Energy savings potential, 2002.

[28] M. Ruta, F. Scioscia, E. D. Sciascio, and G. Loseto. Semantic-based enhancement of ISO/IEC 14543-3 EIB/KNX standard for building automation. *IEEE Transactions on Industrial Informatics*, 7(4):731–739, 2011.

[29] R. Schlegel, S. Obermeier, and J. Schneider. Structured system threat modeling and mitigation analysis for industrial automation systems. In *Proc. of the IEEE 13th Int. Conf. on Industrial Informatics*, pages 197–203, 2015.