SMDAps: A Specification-based Misbehavior Detection System for Implantable Devices in Artificial Pancreas System

Philip Virgil Astillo[1], Jaemin Jeong[1], Wei-Che Chien[2], Bonam Kim[1], JoungSoon Jang[3], Ilsun You[1]

[1] Department of Information Security, Soonchunhyang University, South Korea
[2] Department of Computer Science and Information Engineering, National Dong Hwa University, Taiwan
[3] Department of Internal Medicine, Chung-Ang University College of Medicine, South Korea
pvbastillo@gmail.com, woals9179@naver.com, wcc@gms.ndhu.edu.tw, kimbona9@gmail.com, ilsunu@gmail.com

Abstract

Implantable medical devices are playing a key role in the paradigm shift of providing healthcare services. Particularly, this paper highlights the role of artificial pancreas system (APS) in the management of blood sugar level, especially to patients that are diagnosed with Diabetes Mellitus (DM). APS provides convenience in the self-management of blood sugar level. However, because of the added wireless connectivity feature, the system can be exposed to more security threats and attacks. Hence, it is essential to resolve the security and privacy issues for APS. In this paper, we first introduce the basic architecture of the existing APS and elaborate the roles of each component. Then the security challenges for APS are discussed starting from the component that poses high risk to the patient's health and safety. To address those challenges, we propose a specification-based misbehavior detection system, called SMDAps, which monitors events within the APS to detect misbehaving components based on the behavior-rule that are derived systematically from the embedded system requirements. Moreover, the monitoring task is supplemented with an outlier detection method to detect anomalous glucose data points. To demonstrate the effectiveness of our approach, we emulate the functionalities of the embedded devices integrated into the APS and adopt a glucose-response model found in the UVa/Padova simulator. Based on investigation, the proposed glucose outlier detection can accurately distinguish anomalous glucose data points of more than 94% when such points deviate of more than 5% from the true value. Additionally, the effectiveness of SMDAps showed a dominating detection rate at a considerable degree when compared to the contemporary machine learning approaches such as Support Vector Machine and k-Nearest Neighborhood classifiers. The SMDAps, kNN, and SVM achieve a AUROC of 99.98%, 99.96%, and 99.95%, respectively, for detecting aggressive attacker type associated with the duration of exposure during the simulation runtime.

Keywords: Diabetes, Implantable medical device, Artificial pancreas system, Intrusion Detection System (IDS), Specification-based IDS

1 Introduction

The future of healthcare providers is expected to transform its method in delivering medical services to patients [1-2]. Implantable medical devices (IMD) have currently received growing attention by many researchers to drive the paradigm shift of healthcare services. These devices are placed inside or on the surface of the human body through surgical procedures and are intended to remain there if necessary. Coronary stents, hip implants, intraocular lenses, cardiac pacemakers, cardiac defibrillators, and artificial pancreas system (APS) are some known IMD that many patients use today. These devices are theoretically safe, but those could be dangerous to the human body if proper medical procedures are not followed. Among these devices, we believe that the last three devices mentioned are the most critical one since they are responsible for the vital organs of the human body, thus safe operation is particularly important. Especially, this paper pays more attention to the APS.

As human life spans increase, the prevalence of diabetes is remarkably increasing. According to the World Health Organization, the reported global incidence of diabetes is continuously increasing for the past 3 decades. The number of patients has quadrupled since 1980 and age demographic has widened ranging from youth (below 18) to senior group (above 70) [3]. For diabetes, continuous blood glucose monitoring and accordingly, prompt insulin administration are essential for treatment. Manual management is evidently a laborious task, especially to those that are diagnosed with type 1 DM. For this reason, diabetes patients seek a more sustainable system that can mimic

*Corresponding Author: Ilsun You; E-mail: ilsunu@gmail.com
DOI: 10.3966/160792642021012201001

a healthy pancreas fits to that requirement. APS is a closed-loop control management system that combines insulin pump with continuous glucose monitor to assist the automatic adjustment of hormone insulin delivery. There are several medical device manufacturers, which are racing to develop an APS with the support of Food and Drug Administration (FDA).

An existing model of APS is composed of three basic components: a continuous glucose meter (CGM), a control algorithm platform, and an insulin pump as shown in Figure 1. The collaborative function of these components provides automated regulation of blood glucose level. Accordingly, the burden of the patients on diabetes self-management is minimal. In this model, the CGM periodically provides the person's sugar level to the control algorithm platform, which subsequently computes the appropriate insulin dose and sends command to the subcutaneous insulin pump to deliver the hormone insulin in the human body [4]. Now considering that the different elements in the system communicate wirelessly, it suffers the same security challenges from many wireless systems. Although, much research has been conducted to obtain the optimum insulin dose to ensure health safety, security aspect is still underdeveloped.

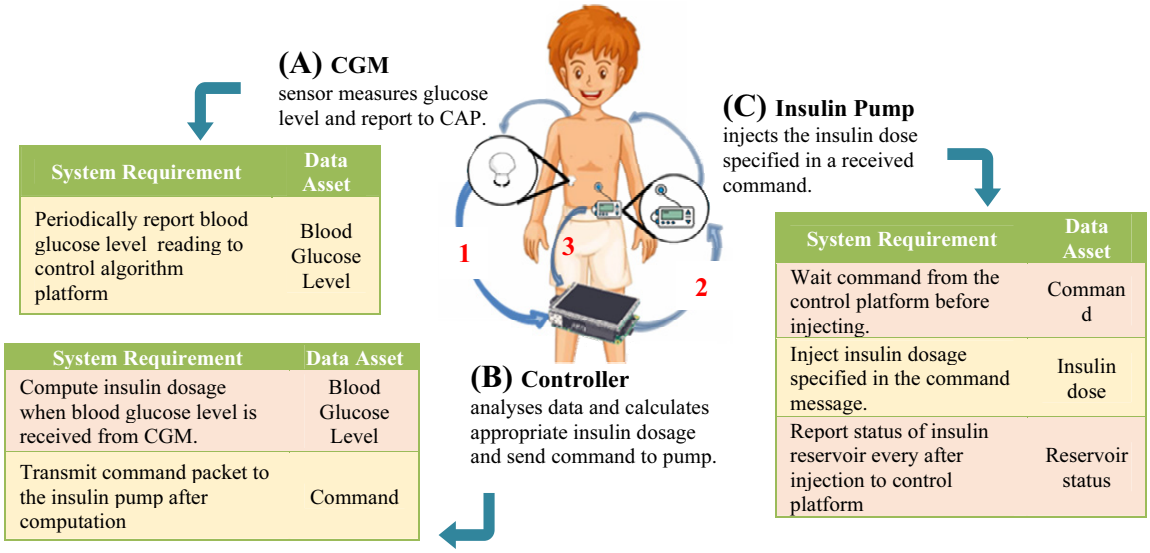


Figure 1. An exemplary of the open-source artificial pancreas model and communication sequence along with the embedded system requirements of each element

Intrusion detection system can be considered as one of the effective security solutions; however, the two representative approaches, i.e., signature-based, and anomaly-based detection, are not suit for APS due to the following reasons. The former cannot address well zero-day attacks, which APS is vulnerable to because of the difficulty to update the system in a timely manner, while the latter can cause APS to suffer from heavy computation overhead because of the dependency to machine learning or statistical profiling [5]. Accordingly, as an effective alternative, we propose a specification-based misbehavior detection, called SMDAps, wherein we specify the intended behavior of APS. To the best of our knowledge, this is the first study that attempts to secure interconnected devices specific to the APS environment through network intrusion detection. The contributions of this paper are the following:

- We analyze the operations of APS and identify the vulnerabilities and security challenges.
- We systematically derived the behavior-rules based on embedded software requirements of each individual devices.
- We modified the Kalman-filter estimation method to assist the monitoring task for glucose outlier detection.
- To evaluate our approach, we extended the FDA-approved T1DM simulator called UVa/Padova by adding the communication functionalities of each devices while maintaining the quality of the glucose response model.

The remainder of this paper is organized as follows. Section 2 discusses the security challenges of APS based on open-source model and in turn express our motivations in response to those challenges. Section 3 surveys some existing approaches of misbehavior detection. The proposed solution to address the identified problems is presented in Section 4, followed by the experimental results in Section 5. Section 6 finally concludes this paper.

2 Security Challenges and Motivations

Highlighting the security issues of each module in APS is imperative. According, we analyzed and present in this section the threats of existing APS

model start from the module that poses high risk to patient's safety.

2.1 Subcutaneous Insulin Pump

The improvements of insulin pump made the device suitable for everyday use in delivering insulin to the body and much portable compared to their predecessors. Its latest added feature is the wireless connectivity, which unfortunately causes patients to be faced with increase threats and attacks [4]. An adversary could be able to remotely change device settings that places the life of patients in danger, i.e., causing hypoglycemia if the pump dispenses more than the recommended insulin dose [6]. Hence, this device can be accounted with highest risk to patient's health if not quickly mitigated.

2.2 Control Algorithm Platform

The control algorithm is the one responsible for computing the appropriate dosage of insulin, with respect to the received glucose level, that the pump module should deliver to the human body. When this module transmits a dispense command, the pump device acts accordingly. Currently, the personal computer (PC) or smart phones are the most common platform where the control algorithm is deployed. Multiple incidents showed that these platforms are vulnerable to malware such as viruses, worm, etc. [7], and this presents threats to the AP system. If compromised, it may send malicious commands to the insulin pump.

2.3 Continuous Glucose Monitor (CGM)

CGM is weighed with lowest risk in the preservation of human lives since it only reads patient's body physiological condition. However, this module is accounted to be an obstacle in achieving the safety requirement of the system because it may operate maliciously if compromised by providing incorrect blood glucose level [8]. In addition, current CGMs are also equipped with wireless connectivity to transmit the patient's condition (glucose level) to a trusted component. Thus, it also suffers from privacy issues.

Outside attackers that impersonates either of these components are also a major threat for they can mislead the legitimate ones in the system. Even though no actual attack has been reported by patients or manufacturers, the author in [9] and [10] used laboratory experiments to prove that such security issues exist on commercial devices. The safety of patients is crucial if these vulnerabilities are exploited by malicious hackers; hence, it is imperative that these devices are protected against hostile attackers. The FDA has already encouraged insulin pump manufacturers to address security in their products. In response, the manufacturers started to employ contemporary security measures, i.e., lightweight encryption and authentication, on their latest products as an implicit solution to secure the interconnected devices. Meanwhile, despite the efforts of the FDA in warning the public on using the old models of commercial products [11], more and more individuals are joining in a subcultural diabetes community who hacked their own insulin pump and develop a do-it-yourself APS using an open-source code such Open-APS or AndroidAPS [12]. Their decision in joining is due to financial cost as well as it allows them to customize the system according to their personal needs. This motivated us to develop an intrusion detection system based on specification-based approach that caters not only to the new products but more importantly to the older ones. Moreover, we believe that contemporary measures are not enough to protect the system from attacks because adversary can always find ways to bypass with it.

3 Existing Approaches

Misbehavior detection method is an effective solution to mitigate the risk of network attacks [13]. To-date, signature-based, anomaly-based, and specification-based method are known types of misbehavior detection techniques. The signature-based method is weak in detecting unknown threats or zero-day attacks for it heavily relies on the patterns of known threats [14]. The weakness of this method makes it unsuitable solution in the APS environment because of the difficulty in quickly updating the system if new vulnerabilities arise. Meanwhile, anomaly-based and specification-based techniques are effective alternatives in detecting unknown threats because their detection mechanisms depend on the normal operation of the system [15]. Table 1 shows the summary of some current works that employed the said techniques. The former utilizes machine learning (ML) methods like Deep Learning (DL) [16], Support Vector Machine (SVM) [17], and k-Nearest Neighborhood (kNN) [18] to establish a profile of the system's normal behavior from a large amount of collected data. On the other hand, the specification-based approach only requires the derivation of rules (behavior-rules) that are based on the behavioral-specification of the system. Even though both techniques are effective, the latter is a more suitable solution to the APS environment considering that the devices integrated into it have limited resources. The ML techniques are known to have a relatively high computational operation. Related works conducted by [19] and [20] shows the feasibility of specification-based technique in resource constrained devices. Both works presented a high detection accuracy with low memory consumption, runtime, and computation overhead. However, the derived rules are application-specific (military UAV and PCA), and this cannot be applied to APS. Additionally, the validation of the operational

Table 1. Related works of misbehavior detection in Internet-of-Things

Ref.	Approach	Short Description	Application Domain	Detection Rate
[16]	Anomaly-based IDS	The authors applied Feed-Forward Deep Neural Network to detect Blackhole, Sinkhole, Wormhole, DDoS, and Opportunistic Service attacks in smart home environment	Smart Home IoT	98.0%
[17]	Anomaly-based IDS	The authors integrated four supervised machine learning techniques such as SVM, Decision Tree, Random Forest, and kNN to detect malicious traffic flow in a personal medical device.	Personal Medical Device	98.0%
[18]	Anomaly-based IDS	The authors proposed an intrusion detection framework while utilizing Artificial Neural Network Decision Tree, Random Forest, and kNN to monitor malicious traffic flow and operational data in multiple smart medical devices.	Smart Healthcare system	91.0%
[19]	Specification-based IDS	The authors applied specification-based approach in military unmanned aerial vehicle (UAV) network environment. Derived a behavior-rules are specific to uav operations.	Unmanned Aerial Vehicle IoT (Military UAV)	97.8%
[20]	Specification-based IDS	The authors applied specification-based approach in Patient-Controlled Analgesia (PCA) Device. Derived behavior-rules are specific to PCA.	Medical IoT (Patient controlled analgesia)	>99.0%

data that are transmitted by the monitored devices were not considered in both works. Motivated by this, we adopt the specification-based approach wherein we derive the behavior-rules for APS and supplement it with outlier detection algorithm to validate the integrity of the operational data (blood glucose). In addition, we initially benchmark our current work against the SVM and kNN algorithms, with the intent to show its effectiveness. These algorithms were selected because of its relatively smaller memory and computation requisite compared to DL.

4 Specification-based Misbehavior Detection System

In embedded software development, the functional operation of specific system or device is usually documented extensively in the embedded system requirements specification. Accordingly, it can serve a purpose of not only as reference for the embedded software developers but also as a guide for a more systematic derivation of behavior-rules. This section presents the steps taken for the development of SMDAps.

4.1 APS Behavior-Rule Derivation

Behavior-rules are the foundation of the software agent to determine the state of the trusted components in a system. In this paper, derivation of behavior-rules is based on the model of APS illustrated in Figure 1 together with their assumed embedded system requirements.

Prior to the derivation of the behavior-rules, we enforce a security context to each system requirement to lay-out a strict sense as to how a component in the APS should behave or operate during its lifetime from the viewpoint of the misbehavior detection agent. Consequently, the enforcement of such context results to the formulation of the APS' security requirements. Furthermore, while forming each security requirements, all possible threats that could prevent the system from achieving a corresponding requirement are identified regardless of the criticality and whether such a threat is exploitable or not. Identifying the threats helps in the derivation of a more realistic and meaningful behavior-rules together with the guidance of the CIA triad of confidentiality, integrity, and availability. Accordingly, the behavior-rules serve as the basis for the development of SMDAPs software agent while following the general software development cycle. Figure 2 illustrates the sequential steps leading to the derivation of the APS' behavior-rules.

4.2 SMDAps Software Design

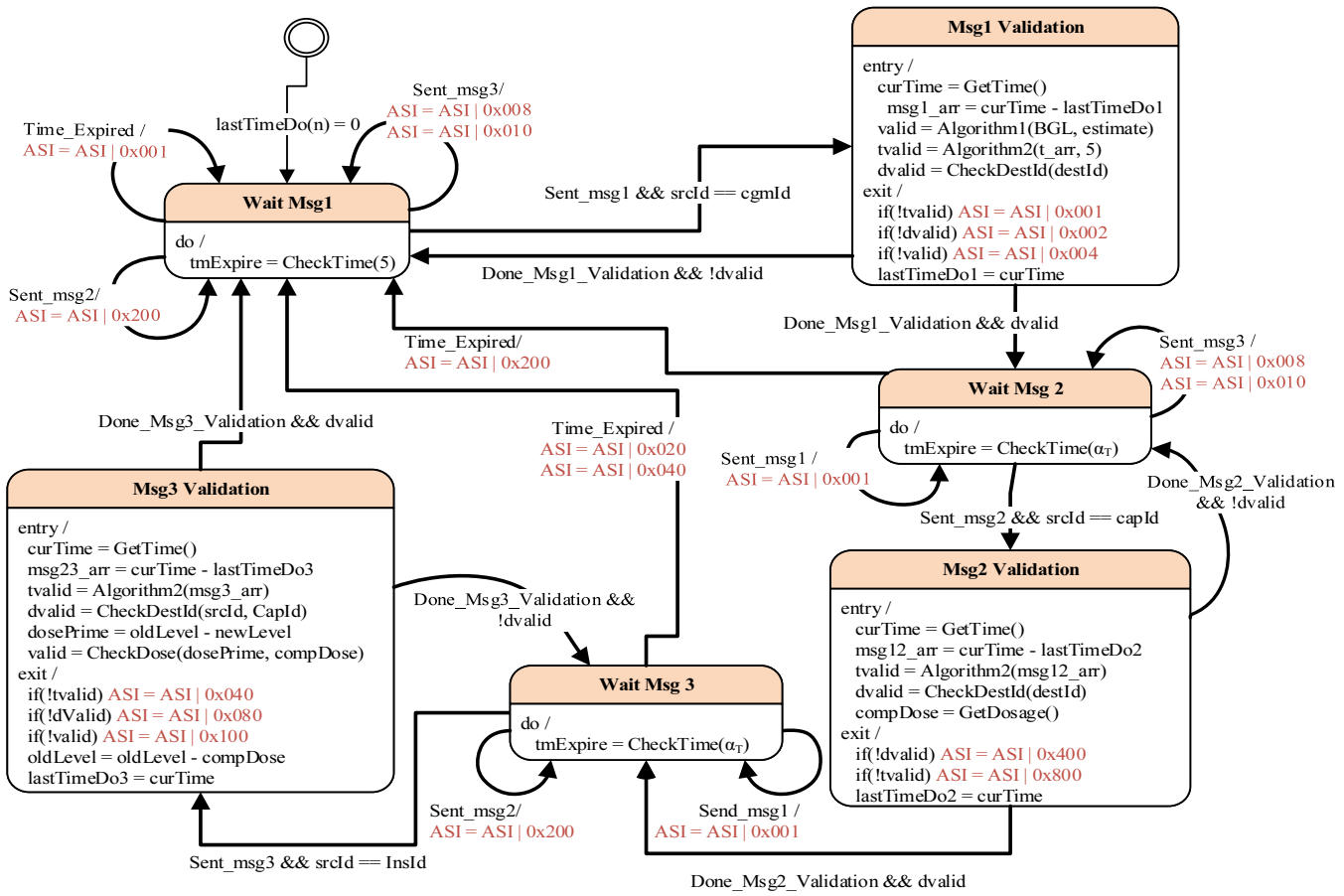
In our proposed approach, we take the behavior-rules as the embodiment of a software requirement for the SMDAps software agent. However, a preceding question arises as to where the software agent is going to operation considering that the APS comprises of embedded devices that are theoretically capable of running a detection agent. In this paper, after studying the operation of the adopted APS model, it was observed that the entire operation is centralized on the control platform. For clarity, Figure 1, shows the sequence of events (*highlighted numbers in red color*) in APS. Accordingly, in this current work, the development of SMDAps software agent is limited to the perspective of the control platform with regards to the APS operations. Nevertheless, we believe that it

Comp-	→ Security Requirements	→ Threats	→ Behavior Rules	← Security Aspect	
CGM	**Shall** report **valid** blood glucose level **only to associated** control platform every *t* minutes.	Does not report at all	Packets must be sent every *t* minutes	Availability	MD Agent
		Does not follow reporting periodicity			
		Sends packets to unassociated control platform.	Packet's destination must only be sent to associated platform.	Confidentiality	
		Sends incorrect blood glucose level	Blood glucose level must be within specified range.	Integrity	
Insulin Pump	Shall **continue** waiting command before injecting until it receives one **only from associated** control platform	Proceed injection without any received command	Device must be in halt mode if no command received.	Integrity	
		Proceed injection when received a command from unassociated source	Device must be in halt mode if received command is from invalid source.		
	Shall inject insulin dosage **correctly** as specified in the command message.	Does not inject correct dosage.	Device must deliver correct amount to insulin dose		
	Shall report **correct** status of insulin reservoir every after all injection event **only to associated** control platform	Does not report at all.	Packets must be sent within a specified interval after the transmission of injection command.	Availability	
		Does not report timely.			
		Sends report to unassociated control platform.	Packets' destination must only be sent to associated control platform.	Confidentiality	
		Sends incorrect insulin reservoir status	Available insulin in reservoir must be within a specified range.	Integrity	
Control Platform	**Shall** provide insulin dosage **only** when blood glucose reading is received **from associated** CGM.	Sends command packet even without receiving glucose report from CGM.	Command packet must be available after receiving blood glucose level from associated CGM.	Integrity	
				Availability	
	Shall timely transmit command packet **only to associated** insulin pump.	Sends command packet to unassociated insulin pump.	Command packet must be sent to associated insulin pump	Confidentiality	
		Command packet is not sent timely.	Packets must be sent within a specified interval after reception of CGM report	Availability	

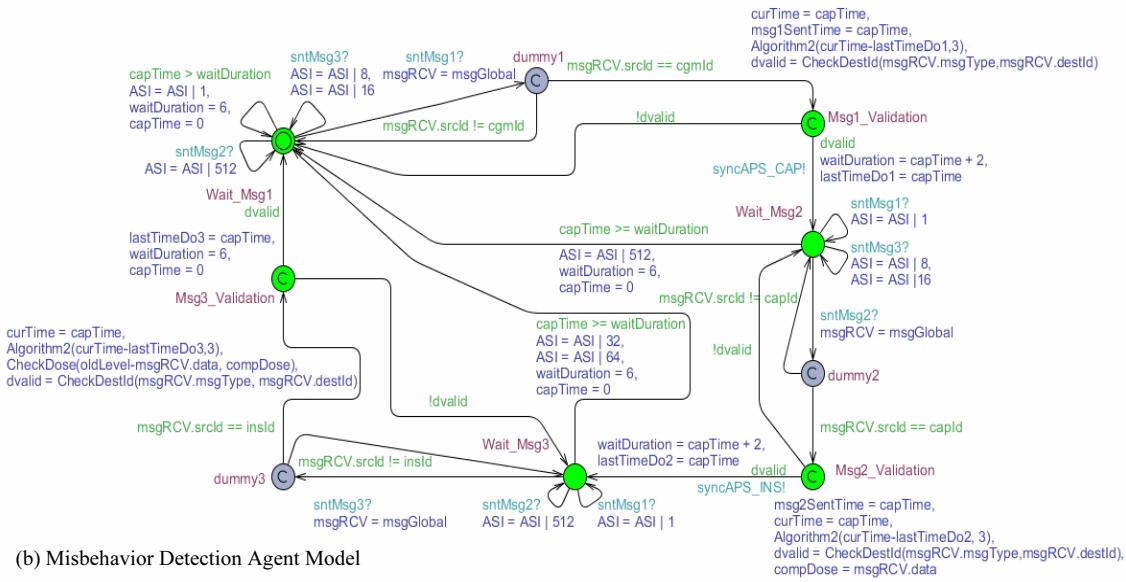
Figure 2. An illustration of the adopted workflow leading to the derivation of the behavior-rules specific to the APS model

would have a safer environment if all three components can monitor each other. This would be one of our future works. Additionally, the deployment of SMDAps agent is limited to platform that does not require connection over the internet. Otherwise, if this requirement cannot be avoided, the monitoring code can be embedded and executed in a trusted platform (e.g. [21-23]) of the controller device. In this way, the monitoring agent will not be compromised even if the main operating kernel is affected by malicious software.

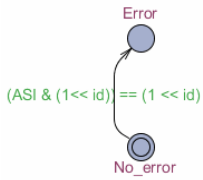
Software models are ways expressing a software design. A commonly used expression is the Unified Modeling Language (UML). As such, this paper employs the UML state diagrams to represent the SMDAps software design. In this case, the drawn UML state diagram technically describe the actions taken by the monitor device in tracking the APS' finite state sequences as mechanism in determining a misbehaving component. Furthermore, this expression is particularly important on the proceeding stage where the functional correctness and completeness of behavior-rules evaluation are verified. AS show in Figure 3(a), the agent enters in 6 states when monitoring the APS condition. The transitions are basically triggered by the corresponding events as well as expiration of time parameter. After modeling the states and transition of the APS' normal operation, supplemental functions, e.g., glucose outlier detection algorithm, are appended on the relevant state. Subsequently, the attack state indicators (ASI) are inserted on the appropriate location and assigned with a coded value, that represents which behavior-rules are violated when an anomalous event occurs. This mean that a value of ASI greater than zero indicates that the system is misbehaving.



(a) UML state diagram



(b) Misbehavior Detection Agent Model



(c) ASI Checker

Summary of queries to the requirement specification (legend: S = safety; R = reachability; N = normal; M = malicious)

ID	Property	Type	Query (CTL)	Satisfied? (N)	Satisfied? (M)
$\boldsymbol{P}_{1=12}$	The behavior-rules 1 ($\boldsymbol{P}_1$) to 12 ($\boldsymbol{P}_{12}$) are eventually violated.	R	E <> Checker(n).Error *where n = 1 to 12*	No	Yes
$\boldsymbol{P}_{2=14}$	The system is deadlock free	S	A[] not deadlock	Yes	Yes

Figure 3. UML State Diagram and UPPAAL model of the MD software agent with summary of queries to the requirement specification

4.3 SMDAps Software Design Verification

While the complete state diagram is formed based on the sequence of APS events and the derived behavior-rules, it is also essential to formally verify the model's functional correctness prior to software

development. Such action adds confidence that the derived behavior-rules are completely covered, otherwise, it will result to high false negative detection rate. In this work, we formally verify the formulated state diagram using an integrated environment tool for system modeling, simulation and verification called UPPAAL [24]. This model check performs state-space exploration which enable the users to assess required specifications by defining it as a computation tree logic (CTL). For more details of this tool, we refer the readers to the most cited tutorial document in [25]. Figure 3(b) shows the equivalent UPPAAL model of our formulated UML diagram. Additionally, it summarizes the verification results of our target properties. From the result, it indicates that the process is safe from deadlocked during runtime. In addition, the reachability property P_1 to P_{12} are also satisfied when misbehaving components are simulated. These results can be an indicative proof of functional correctness for the monitoring tasks and the completeness of behavior-rule assessment.

4.4 Supplemental Algorithms

CGM device periodically transmits glucose reading to the control platform and use it to compute the optimum insulin dosage to be injected by the insulin pump. The CGM device, as demonstrated by [9] and [10], suffers from tampering or impersonation attacks where an adversary provides a malicious glucose level, which consequently lead the control platform to provide an incorrect dosage that can cause an adverse effect to patient's health. To validate these data, we supplement the monitoring task with anomaly detection algorithm, described in Table 2, to detect glucose outlier (Algorithm 1 and Algorithm 2) as well as anomalous transmission time (Algorithm 2). Algorithm 1 utilizes Kalman-filter estimation, Mahalanobis distance method, and sigmoid function to estimate the instantaneous glucose value and followed by Algorithm 2 to classify received data as outlier or not. This estimation method is utilized because it uses a relatively simple operation and does not require high computation power [26]. Note that the Kalman-filter equations are customized to fit the adopted glucose-insulin model. The proposed Kalman-filter based glucose estimation for outlier detection is limited to the glucose-response model of virtual patient under test. This means that the control parameter of Kalman-filter must be replace with the measurable factors pertaining to the physiological sensitivity of patient. Additionally, the adaptation of this method relies on a sensible presumption that APS operates normally at early stage of operation and malicious event exhibit at a later time.

Meanwhile, Algorithm 2 basically provides statistical inference of the underlying distribution of the parameter-of-interest, i.e., Mahalanobis distance error or inter-arrival time of messages, as basis for the classification of events as anomalous or normal.

Table 2. Supplement algorithms for monitoring task

Algorithm 1. Glucose Outlier Detection

Input: glucose reading, previous estimate, previous mahalanobis distance $ME_{i=1}$

Output: current estimate, updated co-variance and measurement noise parameters Rv, and ME_i

(1) Prediction phase of Kalman-Filter method.
(2) Compute ME_i between current reading and predicted estimate from step 1.
(3) Update Rv based on sigmoid function of ME_i
(4) Compute the rate of mahalanobis error change ΔME_i
(5) Perform Algorithm 2 while passing ΔME_i
(6) If valid: Assign Rv_temp to Rv
Else: Assign Rv_temp to ME_i.
Do step 6 then recompute ME_i using estimated Value from step 1 and step 6.
(7) Correction phase of Kalman-Filter method using *Rv_temp* value for measurement noise parameter in Kalman gain computation.

Algorithm 2. Statistical analysis of subject's data distribution

Input: the number of data $Nd_{i=1}$, sum of data $Td_{i=1}$ and mean squared error $TEd_{i=1}$, and current data d_i

Output: Validity (True or False), Nd, Td, TEd_i

(1) Compute mean μ_i and standard deviation σ_i
(2) Compute limiting criterion $LM = \mu_i + \sigma_i + pt$ where pt is called performance control parameter.
(3) If $d_i <= LM$: Update Nd, Td, TEd
return $True$
Else: Retain Nd, Td, TEd
return $False$

5 Experimentation and Evaluation

5.1 Experiment Setup

To demonstrate the effectiveness of our proposed approach, we emulate the behavior of the three components using Raspberry-Pis. In addition, we adopted the glucose-insulin model of an FD-approved Type-1 Diabetes simulator called UVa/Padova Simulator (2008 version). In reference to the open-source python version in [27], we extended the tool in such a way that it will allow us to simulate glucose response to insulin injection and meal in-take of available 30 virtual model patients as well as emulate the actual message exchange between the different components. During the experiment, we also simulate five attacking modes as extremely reckless, reckless, random, cautious, and extremely cautious. The decision when to act maliciously is based on whether a randomly generated number in [0, 1] is less than a threshold value of 0.9, 0.7, 0.5, 0.3, and 0.1,

respectively. This means that the corresponding compromised components behave maliciously at 90%, 70%, 50%, 30%, and 10% of the simulation time. Note that the attack actions include sending anomalous data and violation of message transmission periodicity.

To have a fair comparison between the proposed work with SVM and kNN, the physical variables from the virtual patient, such as the glucose reading from CGM (X_1), estimated glucose value (X_2), the compute Mahalanobis distance error (X_3) and rate of change (X_4), inter-arrival time of messages (X_5) and its error with expected time arrival (X_6), and insulin level status (X_7), were collected at every evaluation cycle during runtime. The collected data comes from both normal state and malicious state in a 90% attack mode. These data serve as the feature vector, in a form of (X_1, X_2, X_3, X_4, X_5, X_6, X_7, y) where y is the classification, for the training of the SVM and kNN classifiers. In the testing phase, another set of data from different virtual patient, which was also collected at runtime, were used. The classification output to each sample is treated as an instance for the computation of the system's compliance degree.

5.2 Effectiveness of Outlier Algorithm

We first evaluate the effectiveness of the proposed outlier algorithms in detecting anomalous glucose data. In real world, anomalous data are effect of data tampering, CGM impersonation or sensor faults. In doing this, we deliberately manipulate at random points the glucose data within 1% to 10% deviation from the true value.

Figure 4(a) shows an example of glucose response from one virtual patient. The graph constitutes of the real data points (*red-star markers*), manipulated data at random locations (*blue-triangle and pink-square markers*) and the estimated glucose value from Kalman-filter method (*light blue line*). Moreover, Figure 4(b) illustrates the distribution of the rate of Mahalanobis error change (ΔME_t) between adjacent data points. ΔME_t corresponds to d_t parameter in Algorithm 2. We can see in the figure that the ΔME_t of manipulated data points falls in the area beyond the limiting criterion. Consequently, the algorithm classifies it as an outlier. Furthermore, based on our investigation, the accuracy of correctly classifying an

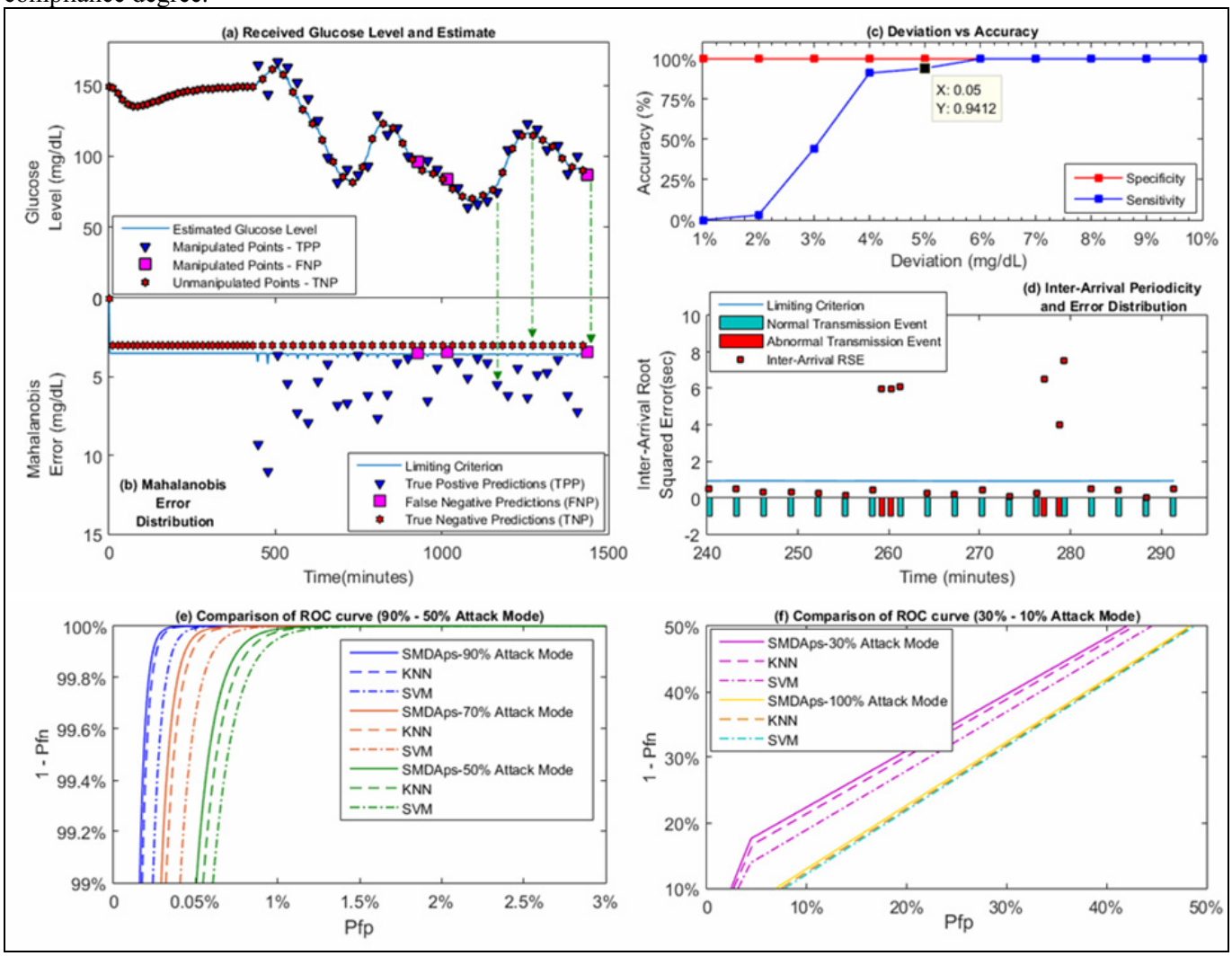


Figure 4. Experiment results; (a) Received glucose level and estimate; (b) Mahalanobis error distribution; (c) Inter-arrival periodicity and error distribution; (d) Deviation vs Accuracy; (e-f) ROC of the different attack mode

outlier, known as sensitivity, increases as the deviation of manipulated data with respect to the true value increases and maintain a low false alarm (1 – *specificity*) close to zero as depicted in Figure 4(c). The algorithm can classify outlier with accuracy of greater than 94% when the deviation is greater than 5% from the true value.

In addition, Figure 4(d) illustrates the inter-arrival periodicity of normal and abnormal message transmission time. Abnormal message transmission time relates to the incorrect reception interval between adjacent messages. This malicious event can also be associated to flooding attacks, tampering, and impersonation. When a message transmission deviates from the specified periodicity, the inter-arrival time value falls beyond the limiting criterion, hence it can also accurately detect that malicious transmission time occur.

5.3 Performance of Misbehavior Detection

In our approach, we set the detection agent to probe the value of the ASI at a uniform discrete-time interval within an aggregation period *Ap.* At the end of every aggregation time space *Ap,* the agent computes the compliance degree of the system by tracking the number of times that the system is in well-behave state, i.e., ASI is equal to zero, and dividing it by the number of probing instances Consequently, at the end of *mth* aggregation period, a compliance degree history c_1, c_2, c_3, ..., c_m is collected and will be used for final classification of the system's state. We adopt the classification approach from [20] which is based on binary grading of the average compliance degree against a minimum threshold value C_T. The average compliance degree is computed based on beta probability distribution function with alpha parameter equal to 1 and parameterized beta using the compliance history and maximum likelihood estimation.

We measure the performance of our approach based on (a) detection rate (1 – *Pfn*)*:* the probability of correctly detecting malicious event; (b) false negative probability (*Pfn*); (c) false positive probability (*Pfp*); and (d) Area under a receiver operating characteristic curve (AUROC) with detection rate vs false positive probability. Figure 4(e) and Figure 4(f) show the effect of different attacking mode on detection accuracy of malicious event in APS. The attackers in Figure 4€ are considered as an aggressive adversary that intend to adversely affect the health of the target patients. Thus, it is imperative that such attackers are accurately detected for immediate mitigation. According, by setting the optimum compliance threshold C_T, we see in Figure 4(e) the ROC curves for attacker, that exposes itself at 90%, 70%, or 50% of the simulation period, has an AUROC close to 100%, P_{fn} *and* P_{fp} less the 1% probability. Specifically, SMDAps achieved an AUROC (*solid lines*) of 99.98%, 99.94%, and 99.90%, respectively. This indicates that the SMDAps can achieve high detection for these aggressive attackers.

On the other hand, a poor performance can be observed in Figure 4(f) when malicious events happened at most 30% of the simulation time. In this case, the attacker exposes itself carefully to avoid detection. Consequently, the AUROC is not more than 56.59% which indicate the weakness of SMDAps in detecting such smarter attackers. Although these adversaries initiate attacks for only a short period of time, their effects cannot be ignored in the context of APS. Hence, it will be considered as another area for improvement in the future works.

Moreover, the Figure 4(e) and Figure 4(f) also display the ROC curves for kNN (*dash lines*) and SVM (*dash-dot lines*). These two machine-learning-based detection techniques also achieved high accuracy in detecting the aggressive type. kNN achieved and AUROC of 99.96%, 99.93%, and 99.87% while SVM obtained 99.5%, 99.91%, and 99.87%. Additionally, both are also weak in detecting smarter attackers achieving an AUROC of not more than 56.0% in kNN and 5476% in SVM. Despite the similarity in the accuracy trendline, AUROC of SMDAps prevails at considerable degree that of kNN and SVM. This is due to inherent imprecision of SVM to learn the optimal hyperplane and finding the optimal k value for kNN. Table 3 summarizes the AUROC comparison of SMDAps, SVM, and kNN. Moreover, we also attribute the edge of SMDAps against SVM and kNN in terms of pre-deployment processes. While SMDAps only requires the derivation of rules, the SVM and kNN must go through collection of necessary features and learn from them the behavioral pattern of the system.

Table 3. Summary of AUROC

Attack Mode	Method		
	SVM	kNN	Our Approach
90%	99.95%	99.96%	99.98%
70%	99.91%	99.92%	99.94%
50%	99.87%	99.88%	99.90%
30%	54.76%	56.05%	56.59%
10%	51.14%	51.32%	51.59%

5 Conclusion

In this paper, we first study the operations of the artificial pancreas system and discuss the security challenges of each component starting with the device that poses highest risk to patient's safety. To mitigate the security and safety threats, we propose a specification-based misbehavior detection system called SMDAps. The misbehavior detection mechanism of SMDAps is based on a systematically derived behavior-rules of APS environment and assisted with glucose outlier detection method. We demonstrate the feasibility and effectiveness of SMDAps by extending

the FDA-approved UVa/Padova simulator to include actual message exchange while maintaining the quality of the glucose response model. Based on the simulation results, our proposed approach can dominate detection accuracy against SVM and kNN machine learning classifiers.

In the future, we plan to extend our work by adding appropriate weights on the behavior-rules according to criticality and explore the use of fuzzy logic to come-up with a crisp classification of the system's state, e.g., malicious, suspicious, and benign. In this way, the patients or medical professionals can act appropriately to mitigate the situation.

Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (NRF-2020R1I1A2073603) as well as the Soonchunhyang University Research Fund.

References

[1] J.-W. Lo, C.-Y. Wu, and S.-F. Chiou, A Lightweight Authentication and Key Agreement Scheme for Telecare Medicine Information System, *Journal of Internet Technoly*, Vol. 21, No. 1, pp. 263-272, January, 2020.

[2] T. Charrad, K. Nouira, and A. Ferchichi, ECG patch monitor: a telemedicine system for remote monitoring and assisting patients during a heart attack, *International Journal of Ad Hoc and Ubiquitous Computing*, Vol. 34, No. 1, pp. 25-34, May, 2020.

[3] W. H. Oganization, "Diabete," 2020. https://www.who.int/news-room/fact-sheets/detail/diabetes (accessed Jan. 05, 2021).

[4] C. Bresch, D. Hély, S. Chollet, and R. Lysecky, SecPump: A Connected Open Source Infusion Pump for Security Research Purposes, *IEEE Embedded Systems Letters*, pp. 1-4, March, 2020.

[5] V. Korzhuk, A. Groznykh, A. Menshikov, and M. Strecker, Identification of attacks against wireless sensor networks based on behaviour analysis, *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications,* Vol. 10, No. 2, pp. 1-21, June, 2019.

[6] A. Alotaibi, R. Al Khalifah, and K. McAssey, The efficacy and safety of insulin pump therapy with predictive low glucose suspend feature in decreasing hypoglycemia in children with type 1 diabetes mellitus: A systematic review and meta-analysis, *Pediatric. Diabetes*, Vol. 21, No. 7, pp. 1256-1267, November, 2020.

[7] A. Darki and M. Faloutsos, RIoTMAN: a systematic analysis of IoT malware behavior, *16th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '20)*, Barcelona, Spain, 2020, pp. 169-182.

[8] S. Rizvi, R. Pipetti, N. McIntyre, J. Todd, and I. Williams, Threat model for securing internet of things (IoT) network at device-level, *Internet of Things*, Vol. 11, p. 100240, September, 2020.

[9] D. J. Cooke, K. Garcia, A. Guzman, L. Kim, B. Mesia, J. Palmer, S. Shields, and M. Zanussi, Vulnerabilities of the Artificial Pancreas System and Proposed Cryptographic Solutions, Undergrduate Research Showcase, Boise State University, Idaho, USA, April, 2020.

[10] C. Li, A. Raghunathan, and N. K. Jha, Hijacking an insulin pump: Security attacks and defenses for a diabetes therapy system, *2011 IEEE 13th International Conference on e-Health Networking, Applications and Services*, Columbia, MO, USA, 2011, pp. 150-156.

[11] Certain Medtronic MiniMed Insulin Pumps Have Potential Cybersecurity Risks: FDA Safety Communication, *U.S. Food and Drug Administration*, 2019. https://www.fda.gov/medical-devices/safety-communications/certain-medtronic-minimed-insulin-pumps-have-potential-cybersecurity-risks-fda-safety-communication (accessed Jan. 09, 2021).

[12] A. Melmer, T. Züger, D. M. Lewis, S. Leibrand, C. Stettler, and M. Laimer, Glycaemic control in individuals with type 1 diabetes using an open source artificial pancreas system (OpenAPS), *Diabetes, Obesity, and Metabolism*, Vol. 21, No. 10, pp. 2333-2337, October, 2019.

[13] P. V. Astillo, J. Kim, V. Sharma, and I. You, SGF-MD: Behavior Rule Specification-Based Distributed Misbehavior Detection of Embedded IoT Devices in a Closed-Loop Smart Greenhouse Farming System, *IEEE Access*, Vol. 8, pp. 196235-196252, October, 2020.

[14] Z. S. Malek and B. Trivedi, User Behaviour based Intrusion Detection System Overview, *International Journal for Research in Applied Science and Engineering Technology*, Vol. 6, No. 10, pp. 149-156, October, 2018.

[15] S. Hajiheidari, K. Wakil, M. Badri, and N. J. Navimipour, Intrusion detection systems in the Internet of things: A comprehensive investigation, *Computer Networks*, Vol. 160, pp. 165-191, September, 2019.

[16] R. B. Basnet, R. Shash, C. Johnson, L. Walgren, and T. Doleck, Towards Detecting and Classifying Network Intrusion Traffic Using Deep Learning Frameworks, *Journal of Internet Services and Information Security*, Vol. 9, No. 4, pp. 1-17, November, 2019.

[17] A. I. Newaz, A. K. Sikder, L. Babun, and A. S. Uluagac, HEKA: A Novel Intrusion Detection System for Attacks to Personal Medical Devices, *2020 IEEE Conference on Communications and Network Security (CNS)*, Avignon, France, France, 2020, pp. 1-9.

[18] A. I. Newaz, A. K. Sikder, M. A. Rahman, and A. S. Uluagac, HealthGuard: A Machine Learning-Based Security Framework for Smart Healthcare Systems, *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, Granada, Spain, Spain, 2019, pp. 389-396.

[19] V. Sharma, I. You, K. Yim, R. Chen, and J.-H. Cho, BRIoT: Behavior Rule Specification-Based Misbehavior Detection for IoT-Embedded Cyber-Physical Systems, *IEEE Access*, Vol. 7, pp. 118556-118580, May, 2019.

[20] G. Choudhary, P. V Astillo, I. You, K. Yim, I. Chen, and J. Cho, Lightweight Misbehavior Detection Management of Embedded IoT Devices in Medical Cyber Physical Systems, *IEEE Transactions on Network and Service Management*, Vol. 17, No. 4, pp. 2496-2510, December, 2020.

[21] W. Sun, R. Zhang, W. Lou, and Y. T. Hou, Rearguard: Secure keyword search using trusted hardware, *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, Honolulu, HI, USA, 2018, pp. 801-809.

[22] A. P. Fournaris, K. Lampropoulos, and O. Koufopavlou, Trusted hardware sensors for anomaly detection in critical infrastructure systems, *2018 7th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, Thessaloniki, Greece, 2018, pp. 1-4.

[23] A. P. Fournaris, C. Dimopoulos, K. Lampropoulos, and O. Koufopavlou, Anomaly Detection Trusted Hardware Sensors for Critical Infrastructure Legacy Devices, *Sensors*, Vol. 20, No. 11, p. 3092, May, 2020.

[24] UPPAAL: Integrated tool environment for modeling, validation and verification, http://uppaal.org (accessed Jan. 09, 2021).

[25] G. Behrmann, A. David, and K. G. Larsen, A tutorial on uppaal, in: M. Bernardo, F. Corradini (Eds.), *Formal methods for the design of real-time systems*, LNCS Vol. 3185, Springer, Berlin, Heidelberg, 2004, pp. 200-236.

[26] M. Shuai, K. Xie, G. Chen, X. Ma, and G. Song, A kalman filter based approach for outlier detection in sensor networks, *2008 International Conference on Computer Science and Software Engineering*, Hubei, China, 2008, Vol. 4, pp. 154-157.

[27] J. Xie, Simglucose v0.2.1, 2018. https://github.com/jxx123/simglucose (accessed Dec. 25, 2020).

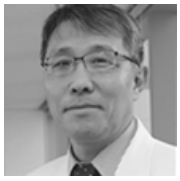
Biographies

Philip Virgil Astillo is a Ph.D. candidate in the Department of Information Security Engineering, Soonchunhyang University, South Korea. He received his M. Eng. degree in computer engineering from the University of San Carlos, Philippines in 2011. His research interests include IoT applications and security, network security, and intrusion detection systems.

Jaemin Jeong received the B.S. degree in Information Security Engineering from Soonchunhyang University, South Korea, where he is currently pursuing the master degree with the Department of Information Security Engineering. His current research interests include mobile Internet security, Internet of Things security, and formal security analysis.

Wei-Che Chien is an Assistant Professor, Department of Computer Science and Information Engineering, National Dong Hwa University, Taiwan. He received Ph.D. degree in the Department of Engineering Science, National Cheng Kung University, Taiwan, in 2020. He His research interests include wireless sensor networks, 5G networks, AIoT, and cloud computing.

Bonam Kim is a research professor, Department of Information Security Engineering, Soonchunhyang University, South Korea. She received Ph.D degree in the department of Computer Science and Software Engeering from Auburn Unversity, USA in 2006. Her main research interests include wireless sensor networks, IoT security, and 5G/6G security.



JoungSoon Jang is professor of Medicine, Department of Internal Medicine, Chung-Ang University College of Medicine Seoul, South Korea. He received M.D. and Ph.D. degree from Hanyang University Seoul, South Korea. His main research interests include clinical trial, wearable medical devices and aviation medicine.

Ilsun You is a university professor, Department of Information Security Engineering, Soonchunhyang University, South Korea. He received his M.S. and Ph.D. degrees in computer science from Dankook University, Seoul, South Korea, in 1997 and 2002, respectively. He received another Ph.D. degree in Kyushu University, Japan, in 2012. His main research interests include 5G/6G security, IoT security, authentication, access control, and formal analysis.