Threat analysis using STRIDE with STAMP/STPA

Tomoko Kaneko
Institute of Information Security
Tokyo Denki University
Tokyo, Japan
knktmk204th@gmail.com

Yuji Takahashi
Tokyo Denki University
Tokyo, Japan
takahashi_yuji@soka.gr.jp

Takao Okubo
Institute of Information Security
Tokyo, Japan
okubo@iisec.ac.jp

Ryoichi Sasaki
Tokyo Denki University
Tokyo, Japan
r.sasaki@mail.dendai.ac.jp

***Abstract*—The safety analysis method called system theoretic accident model and processes (STAMP) and its safety analysis application, system theoretic process analysis (STPA), have attracted much attention as a new safety analysis method for complex Internet-of-Things (IoT) systems. STAMP/STPA is disseminated as a safety analysis technique, but it can also be applied in security risk analysis; a security response of STPA was also presented as STPA-Sec or STPA-SafeSec. This study explores the need for threat analysis from the perspective of cyber-security in STPA-Sec and STPA-SafeSec. Specifically, the STRIDE model is applied to the smart grid case discussed in a previous SafeSec paper, and its effect is evaluated.**

Keywords—STAMP/STPA， Threat Analysis, STRIDE, Safety, Security by Design, Threat Modeling， STPA-Sec， STPA-SafeSec

I. INTRODUCTION

In the Internet-of-Things (IoT) era, developing safer equipment and systems that protect against threats to interconnected systems is necessary. In addition to the confidentiality, integrity, and availability of conventional information security attributes, the perspective of safety is important. Herein, we are using the system theoretic accident model and processes (STAMP) [1] [2] and system theoretic process analysis (STPA) [3] [4] methods to conduct risk analysis from the perspective of safety.

STAMP/STPA has been used for safety, but it can also be applied to security risk analysis [4] through STPA-Sec, a proposed STPA-Security analysis method [5] [6]. STPA-SafeSec, an extension of STPA that integrates and analyzes safety and vulnerability, has also been proposed [7] [8].

However, we believe that comprehensively identifying threats through security using the features of STAMP focusing on interaction requires further ingenuity. Thus, we propose a method of threat analysis using STRIDE with the STAMP model and STPA procedures.

Furthermore, to clarify the features of this method, we determined and verified necessary and sufficient conditions for security applications of the STAMP/STPA safety analysis method.

The necessary conditions for security of STAMP/STPA are as follows:

① safety and security can be analyzed together;

② safety and security are conducted top–down;

③ vulnerabilities and threats can be analyzed together.

The sufficient conditions are as follows:

① vulnerabilities and threat-based risks have been extracted (logic);

② necessary security requirements are obtained in the early stages and incorporated into a security design (validity);

③ combined consideration of safety and security at any stage (exhaustive);

These requirements and sufficient conditions are verified herein.

The remaining paper is organized as follows. Chapter II introduces techniques and concepts related to STAMP, various relevant hazards, and security requirements analysis. In Chapter III, we explain STPA-Sec and STPA-SafeSec security correspondence methods of STPA and their tasks. In Chapter IV, we explain the difference between threat analysis and vulnerability analysis, compare each method, and explain how to apply threat analysis to STPA. In addition, the application of STRIDE analysis to the case of the STPA-SafeSec smart grid is described. Chapter V discusses the proposed method, and Chapter VI summarizes the study and presents future scope.

II. RELATED STUDIES

A. STAMP and related methods

STAMP is an accident model based on systems theory, and STPA is a typical method based on the STAMP model used for hazard analysis. Many system accidents are not caused by component failure, rather they are caused by the interaction of control-related elements (both control and controlled elements) for safety in the system. This mechanism is explained by focusing on the element (component) and the interaction (control action). “The cause for an action not working” is specified by considering that it is equivalent to “the inappropriate control action.”

As a process, STAMP utilizes specifications, safety guide designs, design principles, system engineering, risk management, management principles, and regulation of organizational design. Based on the STAMP model, an accident/event analysis (CAST: causal analysis based on STAMP), hazard analysis (STPA), early concept analysis (Steca: systems-theoretic early concept analysis), systematic/cultural risk analysis, leading indicator identification, and security analysis (STPA-Sec) are presented. An accident/event analysis (CAST) is a method of analyzing an event after an accident has occurred, and STPA-Sec is a security analysis method. STPA-SafeSec has been proposed as a method of integrating safety and security [7] [8].

B. Hazard analysis method

Fault tree analysis(FTA), failure mode and effect analysis (FMEA), and hazard and operability studies (HAZOP) comprise a traditional hazard analysis method that analyze hazard factors by using fault trees and impact analysis tables. The method can be applied beginning with the architecture design stage where the system components and failure modes are determined.

Traditional hazard methods can analyze a single failure of a device or an organization as a hazard factor in a systematic manner by logically forming a branch condition. However, analyzing multiple failures of devices or organizations as complex factor is difficult, which requires an overall field of view, such as an accident generated from the interaction between components.

Based on the STAMP model, STPA is a safety analysis method for analyzing hazard factors using a control structure and control loop diagram, which comprises the components involved in the realization of safety constraints and their interaction with the control structures. STPA can be applied from the conceptual design stage in which the rough components of the system are determined. In a complex system in which multiple devices and organizations (human) interact, we analyze the characteristics of the hazards that lurk in the interaction and use guide words based on past accident case data. One can also analyze the behavior of the entire system.

C. Security by Design

According to NISC (National Information Security Center) of the Cyber Security Center of the Cabinet Office in Japan, “Security by Design” comprises “measures to ensure information security from the planning and Design Stage” [9]. “Secure IoT A general framework for the security of the system” [10] is an important concept that is raised as a basic principle for that purpose.

As the prevalence of IoT and security threats to it have likely caused a great deal of damage, there is a need to ensure security in advance in the early stages of planning and requirements definition processes and design processes (Fig. 1).

Fig.1. Definition of “Security by Design”

D. Methods and types of security analysis

Security analysis techniques include attack trees [11] [12], misuse cases [13], Microsoft Security Development Life Cycle [14], and threat modeling including STRIDE analysis [15]. The security development life cycle [14] provides a detailed data flow diagram and a threat analysis with STRIDE. We extract security requirements in the design stage with an emphasis on ensuring safety by design. A threat tree classification based on STRIDE [15] is also shown. Some HAZOP-based security analysis methods have also been proposed [16].

However, compared with traditional standardized safety analysis methods such as FTA and FMEA, no security analysis method has been standardized. No security design method is widespread in the development field.

As security risks are caused by threats and vulnerability, security analysis is divided into vulnerability and threat analysis. Threats are caused by attackers, and various assets are exploited and threatened by attackers.

III. STPA-SEC AND STPA-SAFESEC IN THE DIRECTION OF SOLVING PROBLEMS

A. Threat and vulnerability analysis

The author believes that STPA-Sec and STPA-SafeSec have some limitations from the perspective of threat analysis based on an attack by a malicious person using a top–down approach. Hence, it is important to show how the addition of threat analysis remedies the limitations of STPA-Sec and STPA-SafeSec.

STPA-Sec[6] currently focuses on the conceptual mission and business level. It is unclear how security engineering will implement the areas it focuses on. STPA-Sec currently focuses on mission business operations and system vulnerabilities. The procedures and cases of systematic threat analysis have not been made public. For unsecure control actions, there is no detailed procedure for deriving security constraints.

At the STPA-SEC stage of Step 2 in the procedure, explaining the necessary and sufficient properties of causal factors regarding security when deriving the scenario of the factor is not possible. The identification of security factors is supposed to be dealt with using additional hint words. However, these hint words only add a partial, poorly shaped information operation to the hint word of hazard factor analysis. There has been no explanation regarding why these additions were made to identify security factors. The authors surmise that more detailed security causal factors (SCF: the Security Causal Factors) are required for threat analysis.

Security analysis can be roughly divided into top–down threat analysis and bottom–up vulnerability analysis. In the vulnerability analysis, vulnerabilities of targeted devices or systems are analyzed and measured; however, vulnerability analysis cannot be implemented unless objects are physically determined. However, threat analysis is a modeling technique and can be analyzed even if the object is at the concept stage. STPA-SafeSec is vulnerable as an example that is likely to be a security-derived hazard causal factor (HCF), and it deals only with vulnerability, without analyzing threats to modeling. To judge whether these will actually become HCF or not, it is expected that the use of threat analysis, a security analysis method for analyzing from the perspective of the attacker, will be used.

Herein, we discuss the effectiveness of threat analysis through a case study, and then apply our threat analysis with a brief explanation of STPA-SafeSec.

B. Method comparison perspectives

The significance of (1)-(4) and its requirements is discussed when comparing the security analysis of STPA-Sec and STPA-SafeSec.

(1) Safety and security framework setting

IoT systems comprise connected things and networks and thus should be regarded as integrated systems of IT (information technology) with physical components. It is important to ensure physical safety in addition to existing information security measures [10]. This requires being

able to analyze safety and security together, and it is hoped that the coverage of safety and security will be considered at any stage.

(2) Security by Design

Security by Design is very important, as security requirements are implemented early in the development phase and the appropriate security features are mounted on the equipment itself. If the wrong equipment is selected for a security feature, it is then too late to think about security measures, and it will backfire.

(3)Modeling against threats

Threat modeling [15] can create a scenario that captures a threat from the perspective of the attack and tie it to the countermeasure. Risk analysis requires that we also be able to analyze threats with vulnerability analysis, and that we can verify the logic by extracting the risk based on vulnerabilities and threats.

(4)Confidentiality

Essential requirements for ensuring users' safety must be determined, as well as the confidentiality of information in IoT systems, including the functions of devices[10].

C. STRIDE with STPA

We propose the procedure for applying STRIDE to STPA as follows. The difference between STPA and STPA-Sec is shown in blue letters and the part added by STRIDE application is shown in red letters.

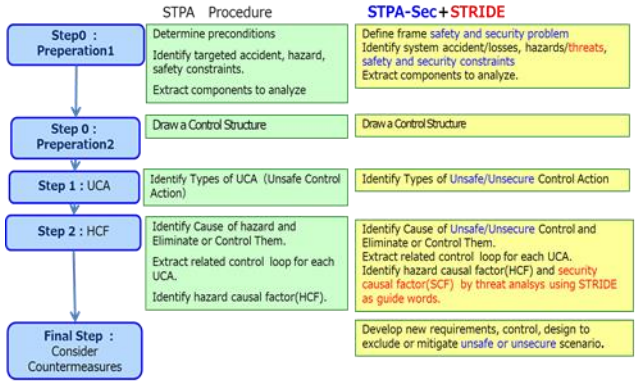

Fig.2 Process of STPA-Sec + STRIDE method

(1)Safety and security framework setting

STPA-Sec introduces a framework that includes hierarchy. Six categories of STRIDE are additionally applied as hint words of security causal analysis in Step 2. Safety and security are handled in parallel on the same Control Structure.

(2) Security by Design

Incidentally, although there is a detailed part added in red in Step 2 in the STPA-Sec procedure, our + STRIDE method does not adopt the procedure. Step 2 of STPA identifies the SCF for the unsecure control action identified in Step 1, but the further detailed procedure is the range of tailoring. The part added in Step 2 of STPA-Sec appears to be effective in problem analysis in the concept phase, but it is not applicable because it is targeted for security design in the + STRIDE method.

(3) Modeling against threats

As a hint word of security factor analysis in Step 2, threat analysis such as that provided by STRIDE, we can add threat modeling. Step 2 is a state after what was a security problem in the previous steps. We identified the assets to be protected as components of the control structure. It is the state after the control action which is that the interaction between the components is insecure . Identifying the unsecure state using the four guide words does not reveal the vulnerability information of the device itself but does identify the unknown threat. We will identify the security factors for this unknown unsecured state (= threat),. The proposed method using STRIDE for the first time in Step 2 makes it easy to distinguish between factors and countermeasures at this stage and details the procedure of threat modeling that STPA does not specify in detail.

In other words, the proposed method includes comprehensive identification of the security risk as an interaction advocated by STPA-Sec.

(4) Confidentiality

Using STRIDE enables analysis based on attributes such as confidentiality other than availability or integrity.

D. Microgrid examples of STPA-SafeSec

The document [7] uses a microgrid as a case study, with the analysis of the connection between a wide-area power network and a local power network.

The procedures for security are considered according to the contents of the case.

Step 0 Preparation 1 (STPA-SafeSec II ~ IV) identifies safety constraints and security constraints for each hazard. We identify the safety and security constraints at a high level of abstraction by taking the negative form of the hazard, with the constraints being safety constraints (Safety, CSTR-S-n) and availability constraints (Availability, CSTR-A-n), We number the constraints according to their attributes, such as integrity constraints (Integrity, CSTR -I-n). In this case, only CSTR-S-5 from the safety constraint CSTR-S-1 (the negative form of H1 to H5) appears, but the availability and integrity constraints are generally handled.

STEP 0 Preparation 2 (STPA-SafeSec V), build with the control structure in the Control layer.

Step 1 Extract UCA(STPA-SafeSec VI ~ IX).

STEP 2a Build a Component layer (STPA-SafeSec X, XI). At the physical level, the speed controller at the functional level is represented by a specific configuration such as analog digital converters and Raspberry Pi, with USB.

(STPA-SafeSec XII) allocate abstract safety and security constraints to Component layer elements.

STEP 2c (STPA-SafeSec XIII) detailed the abstract hazard scenario to the Component layer.

The STPA-SafeSec paper considers the procedure to be divided into the Control and Component layers. The specific case of STRIDE applied to the Control layer is shown in Section3.3. STRIDE applied to the Component layer is shown in Section 3.4.

E. Effect of applying STRIDE on the Control layer

The principal aim of this paper is to demonstrate that security analysis is possible using a threat modeling by

applying STRIDE even in a Control layer on which STPA-SafeSec cannot analyze security.

For this reason, we specifically apply STRIDE to an additional part of the threat analysis that is not specified in the STPA-SafeSec and STPA-Sec processes. As a procedure, the SCF analysis of Step 2 will be added.This shows the flow when it is implemented using STPA-Sec and + STRIDE.

Step 0 Preperation1:System Engineering Foundations

Define the frame safety and security problem

Guarantee a safe and secure power operation. Today, threats of cyberattacks are increasing in grid power operation, including maintenance. There can be many attack elements including terrorism.

Identify losses or accidents following A1-A4 in Table I. STPA-SafeSec can also be used in the same manner.

TABLE I. IDENTIFY LOSSES OR ACCIDENTS

ID	Accidents/Losses
A1	Injury to humans
A2	Damage to power equipment
A3	Damage to end-user equipment
A4	Interruption of power supply to consumer loads

Next, identify hazards (H1-H7) and threats (T1-T3). In our opinion, not only hazards but also threats must be identified in this step because safety hazards represent security threats.

TABLE II. IDENTIFY HAZARDS (H1-H7) AND THREATS (T1-T3)

ID	Hazard
H-1	Out-of-sync reclosure
H-2	Operation of power equipment outside of operational limits
H-3	Violation of power quality metrics
H-4	Inability to achieve synchronization
H-5	Inability to meet local demand
ID	Threat
T-1	Power equipment is destroyed
T-2	Operation of power equipment is deprived of authority
T-3	Control information of device is stolen

Draw a control structure of the Control layer. Fig. 3 shows the example of microgrid used in the STPA-SafeSec paper.

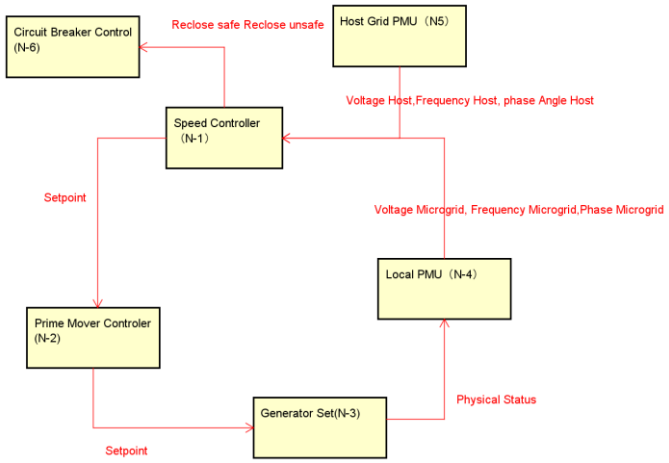

Fig 3. Control structure of Control layer

Step1: Identify Unsafe or Unsecure Control Actions

Unsafe or Unsecure control action from speed controller is shown in Fig. 3 in red. In this case, instructing the prime mover controller to set the value outside the operation range not only leads to hazard (H-2) but also threatens (T-1,T-2). Hence, this control action is unsafe and unsecure. Table III shows 4 types of unsafe and unsecure status for each control action (CA).

TABLE III. UNSAFE OR UNSECURE CONTROL ACTION

No	CA	From	To	Not Providing	Providing causes hazard	Too early or Too late	Stopping too soon or applying too long
1	Reclose safe Reclose unsafe	Speed Controller（N-1）	Circuit Breaker Control(N-6)		The speed controller wrongfully assumes that synchronization is achieved. It would then indicate that the reclosure of the circuit breaker is safe when it is not. (H1)	The speed controller assumes that synchronization is achieved. It would then indicate that the reclosure of the circuit breaker is safe while it is too early or too late.(H1)	
2	Setpoint	Speed Controller（N-1）	Prime Mover Controller (N-2)	When the breaker is in the released state, set values within the operating range are instructed to the prime mover controller with Not（In other words, the setting value is not updated）（H-3,H-4,H-5）	Instructs the prime mover controller to set the value outside the operation range（H-2,T-1,T-2）	When the breaker is in the released state, set values within the operating range are sent as instructions to the prime mover controller with Too late.（In other words, the setting value is not updated）（H-3,H-4,H-5）	
3	Voltage Host, Frequency Host, Phase Angle Host	Host Grid PMU（N5）	Speed Controller（N-1）	Host Grid PMU does not report measured voltage Host, Frequency Host, or Phase angle Host.（H-3,T-3）	Host Grid PMU reports incorrect measured voltage Host, Frequency Host, or Phase angle Host.（H-3,T-1）		
4	Voltage Microgrid, Frequency Microgrid, Phase Microgrid	Local PMU（N-4）	Speed Controller（N-1）	Host Grid PMU do not report measured voltage Host, Frequency Host, Phase angle Host.（H-3,T-3）	Local PMU reports incorrectly measured voltage Host, Frequency Host, or Phase angle Host.（H-3,T-1）		

*PMU: Phasor Measurement Units

Step 2: Identify Causes of Unsafe or Unsecure Control and Eliminate or Control Them

According to STPA-SafeSec [7], security constraints are identified in the Control layer, but security analysis is not performed.

However, if we follow the principle of Security by Design, a security analysis for the speed control (N1) of the Control layer is necessary even in the early stages of development. We also implemented STRIDE analysis, a threat analysis method that can be used in the early phase. In other words, a threat analysis based on an attacker's perspective was conducted to derive the hazard scenario.

Specifically, we attempted to analyze the speed controller in the Control layer using STRIDE. STRIDE is based on a reference architecture for determining the overall image of a system. Enumerating threats as a threat analysis diagram is used to verify mitigation and mitigation measures. The purpose of threat modeling, for which it would be possible to use a layer, was to understand how an attacker could penetrate the system.

It is important to take appropriate mitigation measures and to consider mitigation measures in the early design phase rather than after the system is deployed to eliminate the cost waste. Therefore, in Step 2, we use STRIDE to analyze each attribute and analyze the SCF to ensure that security functions are held as identifiable measures.

In this case study, we focus on the speed controller against the Unsafe or Unsecure CA of "Instructs the prime mover controller to set the value outside the operation range (H-2, T-1, T-2)." STPA-Sec extracts causal factors by adding guide words such as "malformed" or "unauthorized. However, in this study, we decided to use STRIDE as a guide word for this SCF.

The Controller corresponds to the Speed controller in this case. The hint word of STRIDE will be used to refine the process model and control algorithm of the control input, malformed external information, malformed feedback, or the controller itself, which is the input of the following item Controller. To identify the hazard factor as the hint word of STPA instead of STPA-Sec, the hint word specified in STPA is used as it is.

Table IV shows the SCF for the speed controller (CPU). In this case, we extracted STRIDE as a hint word in the threat scenario and took example countermeasures against it. The six classifications of STRIDE represent the required security properties of authentication, integrity, confidentiality, availability, and authorization, as shown in Table IV, and the threats listed from different perspectives show the direction of mitigation for each causal factor. Using a classification Threat Tree [15], which shows the threat mechanism linked to STRIDE, the threat reduction by development and operation is obtained for each threat identified to the point where the attack can occur in STPA. STRIDE has traditionally been used for threat modeling of information systems, but its application to IoT security, including devices that are connected with special applications such as these, has been discussed. Hence, while referring to this commentary in the case of SafeSec, "scenario 1.1: CPT-N1 Speed Controller" recognizes the correct feedback incorrectly. We analyze the SCF in STRIDE. The SCF for the speed controller (CPU) of this case is shown in Table IV, which was extracted from STRIDE as a hint word.

"Repudiation" does not have a corresponding factor in N1-1 because the user does has no means of proving this action. In addition, in Table II, we show the specific threat scenarios and countermeasures for each SCF of N1-1, based on the STRIDE analysis of IoT security [17]. In this analysis, it becomes clear that the attacker's perspective is capable of any kind of attack on the target device. Moreover, each threat is classified by a security property, and hence it is easy to determine the necessary security measures. In addition to the analysis for devices called N1-1, it is possible to guide threat scenarios and countermeasures according to the target, such as communication and storage.

SafeSec analyzes vulnerabilities that could be security threats, but provides no threat analysis from the attacker's perspective, and only availability and integrity are considered as security properties, excluding access control, confidentiality and authentication that results in encryption. This can be accomplished by combining STPA-Sec with STRIDE analysis, which is considered to be an analysis of authenticity that leads to authorization. For the speed controller, one of the components of the control layer, the specific threat scenarios and countermeasures that are assumed for each property could be addressed as a result of the STRIDE analysis shown in Table IV.

F. Effect of applying STRIDE on a Component layer

We discussed how the choice of Component layer can be affected by the STRIDE application of the Control layer speed controller. Raspberry Pi is one of the components of the Component layer. As a result of the STRIDE analysis shown in Table V, we were able to extract the specific threat scenarios and countermeasures for Raspberry Pi that were expected for each property.

TABLE IV. SCF OF N1, THREAT SCENARIOS, AND COUNTERMEASURES FOR THE SPEED CONTROLLER (IN THE CONTROL LAYER)

STRIDE	Required Properties	SCF of N1	Expected threat scenarios	Example of measures
Spoofing identity	Authentication	No correct authentication is made for N1-1 (Speed controller) (N1-S)	Host PMU impersonates the local PMU	Use IC chip with authentication function
Tampering	Integrity	Incorrect FB signal is inserted into N1 (Speed controller) (N1-T)	Some or all of the software running on the speed control is replaced by an attacker	Message authentication Code (MAC), tamper-proof mechanism applied to speed controller
Information Disclosure	Confidentiality	The FB signal of N1-1 (Speed controller) is leaked (N1-I)	If the software running on the speed controller has been modified, the modified software might disclose the plaintext to an unauthorized person.	Implemented anti-malware, Secure Key Management

Denial of Service	Availability	N1 (Speed controller) is destroyed (N1-D)	・The speed controller is exposed to the threat of DoS in the form of constantly waiting for the network for incoming and unsolicited datagrams. ・An attacker can open a large number of connections at the same time and take an extremely long time to process. In some cases, one-sided traffic can undermine the speed controller's ability to handle it. In both cases, the speed controller is virtually a malfunction in the network. -The function of the speed controller stops or cannot communicate by interference or cable cutting.	Limit the number of accesses from an attacker or the same IP. Create a speed controller that can withstand large-scale traffic
Elevation of Privilege	Authorization	(N1-E)	Limit the number of accesses from an attacker or the same IP. Create a speed controller that can withstand large-scale traffic	Access control of the speed controller. Establish an authorization scheme.

TABLE V. SCF OF N1-1, EXPECTED SCENARIO, AND MEASURES OF RASPBERRY PI (IN THE COMPONENT LAYER)

STRIDE	Required Properties	SCF of N1-1	Expected scenarios	Example of measures
Spoofing identity	Authentication	No correct authentication is made to N1-1 (Speed controller CPU) (N1-1-S)	-If the operating system user settings are not set properly, attackers might spoof them	Set the password appropriately: SSH Login with private key
Tampering	Integrity	Incorrect FB signal is inserted into N1-1 (Speed controller CPU) (N1-1-T)	If an illegal program has access to a cryptographic key or an encryption mechanism that holds the cryptographic key, the software replaced will misuse the real ID of the speed controller. An attacker can use the extracted cryptographic keys to intercept, block, and replace data from the speed controller with false data and pass authentication with a stolen cryptographic key.	MAC Applying a tamper-proof mechanism to the speed controller
Repudiation	Accountability	(N1-1-R)	If the Raspberry Pi user does not have a log of the communication, it is likely to negate the fact of the operation that the user performed improperly	Acquisition and maintenance of various logs
Information Disclosure	Confidentiality	The FB signal of N1-1 (Speed controller CPU) is leaked (N1-1-I)	The attacker exploits the encrypted key and obtains the encryption key and decryption key between the speed controller and The Controller (the field gateway or the Cloud gateway), thereby allowing the attacker to get the clear text.	Implemented Anti-malware, Secure Key Management
Denial of Service	Availability	N1-1 (Speed controller CPU) is destroyed (N1-1-D)	The function might be stopped if unauthorized access is performed over a WAN or Ethernet, or when a large amount of data is received.	Apply response limit
Elevation of Privilege	Authorization	(N1-1-E)	If the administrator setting of the OS is not appropriate, the user who does not have administrator rights of the OS originally has administrator privileges, and execution with administrator authority might be used illegally	"Run as Administrator" or "Restrict users who can get administrator rights"

IV. CONSIDERATION

A. *What we learned from the analysis case*

In the case of adding STRIDE analysis in Chapter III, we found the following. The STRIDE analysis is possible in both the Control and Component layers. Although the degree of abstraction is different in the Control and Component layers, each STRIDE threat analysis is possible. Because it is layered, the risk is identified by the Control layer. In addition, it is possible to take measures such as creating a security function.

As a result, it is possible to perform a threat analysis on the components that are more detailed in each component of the Selected Component layer.This will make the security suitable for each stage of the request analysis and the design. This layered insuring of security is itself a "Security by Design."

B. *Comparison of STPA-Sec, STPA-SafeSec and STPA-Sec(+STRIDE)*

The results of comparing the security analyses of STPA-SafeSec, STPA-Sec, and STPA-Sec (+STRIDE), an additional STRIDE method, are described.

(1)Safety and security framework setting

STPA-SafeSec derives system hazard and safety and security constraints, but does not perform security analysis on the Control layer. The Component layer finds a security vulnerability on a more specific physical equipment base for the first time. In addition, STPA-SafeSec deals with two fixed layers the more abstract Control layer and the Component layer of the physical equipment base.

In contrast, STPA-Sec does not differentiate between the analysis stages of safety and security. STPA-Sec simultaneously deals with safety and security. It identifies unsafe or unsecure CAs, identifies causes of unsafe or unsecure control, and eliminates or controls them. STPA-Sec is mostly the same as STPA except for some extensions shown. STPA-Sec incorporates hierarchy similar to a system of systems, which embodies the necessary parts of systems. It is a framework that involves a hierarchical control structure. The steps in STPA-Sec are flexible in layering, and STPA-Sec (+STRIDE) is the same as STPA-Sec.

(2)Security by Design

Considering STPA-SafeSec as Security by Design is difficult because it does not consider security at the top–down from the early stages of planning and requirements definition processes.

In contrast, STPA-Sec adopts a top–down approach. It defines and frames the security problem at the beginning. Identified security requirements in an early stage can be used in the next stage. It is equivalent to Security by Design. STPA-Sec (+STRIDE) is identical to STPA-Sec.

(3) Threat modeling[15]

STPA-SafeSec sets security constraints, allocates them, and measures security vulnerabilities on the Component layer. A security vulnerability is more embodied than STPA-Sec. However, threat modeling is necessary to create a detailed scenario that captures the threat from the perspective of attack. The security analysis in the Component layer of STPA-SafeSec is conducted to analyze the well-known security vulnerabilities of physical equipment. This action is different from threat modeling. STPA-SafeSec does not perform threat modeling.

On the other hand, STPA-Sec contains the problem, vulnerability, and threat analysis. However, STPA-Sec is focused currently on mission business operations and system vulnerabilities. The procedures and cases of systematic threat analysis have not been made public.

However, STPA-Sec （+STRIDE） performs threat modeling. Threat modeling is security analysis from the early stages of the planning and requirements definition processes. It is a method of analyzing by modeling at the stage at which the object is not specifically defined. STPA-Sec（+STRIDE）has a policy of adding STRIDE as a hint word of a security causal factor. STRIDE analysis is a typical threat analysis technique that provides hint words. It shows the required properties and countermeasure examples, with the result that setting up the scenarios and measures to be derived in STEP 2 becomes easy.

(5) Confidentiality

STPA-SafeSec considers safety as well as threats of integrity and availability levels. However, it does not deal with confidentiality, which is considered important for information security. In the case of cyber-physical security, this may also be attributed to the fact that confidentiality is less important than availability and integrity.

Although the security properties of system development in STPA-Sec are not clarified, confidentiality and privacy, such as information leaks, are provided as the accident cases.

C. Evaluation of STPA-Sec, STPA-SafeSec, and the proposed method

Table IV lists the evaluation by comparison from the perspective to be considered when comparing the security analyses of STPA-Sec, STPA-SafeSec, and STPA-Sec (+ STRIDE). The results are assessed based on the requirements of the (1)-(3), and sufficient conditions are shown in Table IV. Because different properties are required for the target system and the product, it is not possible to determine confidentiality as well as safety, availability, and integrity. Therefore, there is no evaluation based on the sufficient conditions required at this time. In summary, STPA-Sec and + STRIDE are more highly rated in terms of (1) safety and security framework setting and (2) Security by Design. (3) In modeling against threats, + STRIDE is more valuable than STPA-SafeSec and STPA-Sec.

In the case of STPA-SafeSec, no security analysis is performed on the speed controller in the Control layer. However, we identified seven concrete SCF, scenarios, and countermeasures using STPA-Sec (+ STRIDE) in the Control layer. In additon, STPA-Sec（+STRIDE） were able to identify six SCFs and more concrete security scenarios and countermeasures for Raspberry Pi in the Component layer.

As selecting the Component layer after applying STRIDE to the speed controller of the Control layer is available, selecting a physical device that can create an appropriate security function that can take countermeasures against the threats derived by the STRIDE analysis is possible. For example, in that case, to take countermeasures against the threat of spoofing to the speed controller, a physical device having a more advanced authentication function must be selected. In that case, there is a possibility that Raspberry Pi is not selected. A more secure and highly functional device will be selected.

TABLE VI. EVALUATION OF STPA-SEC AND STPA-SAFESEC

Methods	① Safety and Security framework setting		② Security by Design		③ Threat modeling	
Condition	necessary condition	sufficient condition	necessary condition	sufficient condition	necessary condition	sufficient condition
Evaluation criteria	① The ability to analyze safety and security together	① Safety and security are analyzed at any stage (coverage)	② Thinking about security from the top down. (Cost reduction)	② Obtaining necessary security requirements in the early stages and make it into security design. (validity)	③ Be able to analyze risks based on vulnerabilities or threats	③Both vulnerabilities and threat-based risks have been extracted (logic)
STPA-SafeSec.	STPA-SafeSec derives system hazards and safety and security constraints, but does not perform security analysis on the Control layer.		It does not consider security at the top down from the early stages of planning and requirements definition processes.		The security analysis in the Component layer of STPA- SafeSec is conductor to analyze the well-known security vulnerabilities of physical equipment.	
Evaluation	○	×	×	×	○	×
STPA-Sec	STPA-Sec does not differentiate between the analysis stage of safety and security.		STPA-Sec has a top–down approach. This contributes to cost reduction. However, this is focused on problem analysis in the concept stage for judging at the business level, and no method for creating security requirements to turn into security functions is presented.		STPA-Sec contains the problem, vulnerability, and threat analysis, but the procedures and cases of systematic threat analysis have not been made public.	

Evaluation	○	○	○	×	○	△ （Details of threat analysis not presented）
STPA-Sec （+STRIDE.）	This is the same as STPA-Sec.		This is a top–down approach that is the same as STPA-Sec. Furthermore, it is possible to present a method of creating security requests by security functions using STRIDE threat analysis .		STPA-Sec（+STRIDE）perform threat modeling in addition to STPA-Sec analysis.	
Evaluation	○	○	○	○	○	○

V. CONCLUSION

A. *Future Work*

Future tasks include quantitative verification of the proposed method as fully as possible. In addition to the case of a microgrid, more detailed security case implementation and application verification, such as IoT security and system operation, are required. In addition, since STPA-Sec is a security risk analysis method, establishing the evaluation method of selection measures and selection criteria is necessary. STPA-Sec recently reported three-layer tailoring cases [18]. It is expected that the process over the life cycle as indicated in the recently published STPA handbook [19] will be confirmed as security engineering.

B. *Summary*

In this study, we propose a method of threat analysis using STRIDE with the STAMP model and STPA procedures. This threat analysis method derives a more comprehensive and practical security scenario from the perspective of threat analysis. STPA and STPA-Sec procedures are better than STPA-SafeSec steps for extracting safety and security risks in early stages of development. In our opinion, it would be better to incorporate the STRIDE threat analysis into the STPA-Sec procedure. After implementing an approach to interaction and non-technical problems, which is the uniqueness of STPA and STPA-Sec, STRIDE is considered to be available for SCF analysis as an additional analysis.

In the future, examining the proposed method is necessary to create more detailed cases and perform quantitative verifications as fully as possible. Safety experts require analytical techniques that can analyze together not only safety but also security to protect the safety of functional and physical systems. Simultaneously, security experts need to analyze system threats and vulnerabilities while considering safety. We expect that a useful technique for both safety and security will be established in future works.

REFERENCES

[1] Nancy G. Leveson, Engineering a Safer World, Systems Thinking Applied to Safety, 2012.

[2] Nancy G. Leveson, *Engineering a Safer World*, MIT Press, 2012.

[3] STPA handbook, http://psas.scripts.mit.edu/home/

[4] Nancy G. Leveson, STAMP Intro and Overview of STPA and CAST, http://psas.scripts.mit.edu/home/

[5] William Young, Nancy G. Leveson. Systems Thinking for Safety and Security, Proceedings of the 29th Annual Computer Security Applications Conference （ACSAC 2013）, pp.1-8 （2013）.

[6] William Young, Reed Porada, System-Theoretic Process Analysis for Security（STPA-SEC）:Cyber Security and STPA, 2017 STAMP Conference

[7] Ivo Friedberg, Kieran, Paul Smith, David Laverty, and Sakir Sezer. STPA-SafeSec: Safety and security analysis for cyber-physical systems, Journal of Information Security and Applications, Volume 34, Part 2, pp.183-196 （2017）.

[8] Keiji Okamoto, Kouzou Okano. STAMP Introduction of overseas cases：STPA-SafeSec, SEC journal 52(in Japanese)

[9] NISC, www.nisc.go.jp/conference/seisaku/dai15/pdf/15siryou02.pdf

[10] NISC, General Framework for Secure IoT Systems, https://www.nisc.go.jp/eng/pdf/iot_framework2016_eng.pdf

[11] Bruce Schneier, Attack Trees. Dr. Dobb's Journal of Software Tools 24(12) (1999) 21–29.

[12] Barbara Kordy, Sjouke Mauw, Saša Radomirović, Patrick Schweitzer, Foundations of Attack–Defense Trees, FAST 2010: Formal Aspects of Security and Trust pp 80–95.

[13] Guttorm Sindre and Andreas L. Opdahl, Eliciting security requirements with misuse cases, Requirements Engineering, Vol.10, No.1, pp. 34–44 (2005).

[14] Steve Lipner, Michael Howard, The Trustworthy Computing Security Development Lifecycle, https://msdn.microsoft.com/en-us/library/ms995349.aspx

[15] Adam Shostack, Threat Modeling: Designing for Security, Wiley 2014.

[16] Jingxuan Wei, Yutaka, Matsubara, Hiroaki Takada, HAZOP-Based Security Analysis for Embedded Systems: Case Study of Open Source Immobilizer Protocol Stack.

[17] Microsoft, Azure, Internet of Things security architecture, https://docs.microsoft.com/ja-jp/azure/iot-accelerators/iot-security-architecture

[18] Captain Martin Span, Major Logan Mailloux, and Colonel William Young. STPA-Sec Case Study of a Next Generation Refueling Aircraft, Air Force Institute of Technology and Eglin Air Force Base,http://psas.scripts.mit.edu/home/2018-stamp-workshop-presentations/

[19] STPA Handbook, http://psas.scripts.mit.edu/home/get_file.php?name=STPA_handbook.pdf