ABSTRACT

Secure system design faces many risks such as information leakage and denial of service. We propose a method named CC-Case to describe security assurance cases based on the security structures and threat analysis. CC-Case uses Common Criteria (ISO/IEC15408). While the scope of CC-Case mainly focuses on the requirement stage, CC-Case can handle the life-cycle process of system design that contains the requirement, design, implementation, test, and the maintenance stages. It can make countermeasure easily against the situation which an unexpected new threat produced by invisible attackers incessantly.

KEYWORDS


1 INTRODUCTION

Customers expect that IT products and systems satisfy the necessary conditions not to fall into any dangerous situations. Developers must build up IT products and systems avoiding many risks. Then risk management is important to support the development. We face many risks at every stage of system development such as requirement analysis, design, development, test and service provision stage. It is really important to assure the countermeasure against risks in the each process of system development.

It is important to handle security risks because security accidents have serious influences. However, there are no established methods to assure the validity of security risk up to this time. In this paper, we propose a security assurance case method against risks. This assurance case makes clear the elements to assure against system risks, and the process to argue with a customer and to get an agreement. It gives the way to make systems trustworthy by effective arguments with stakeholders.

In Chapter 2, we explain assurance case, security assurance case and risk management. In Chapter 3, we show significance of assurance case, current system risks and handling by assurance case, difficulty of treatment for security risks. Chapter 4, we show the concept of proposed method and its provision to the life-cycle process. In Chapter 5, we explain example of concrete model and the merits of CC-Case. In Chapter 6, we show summary and future tasks.

2 RELATED STUDIES

2.1 Assurance Case

Assurance case, which is defined in ISO/IEC15026 part2, is a method for describing a system’s critical security level. Standards are proposed by ISO/IEC15026 [1] and OMG’s Argument Metamodell (ARM) and [2] Software
Assurance Evidence Metamodel (SAEM) [3]. ISO/IEC 15026 specifies scopes, adaptability, application, assurance case’s structure and contents, and deliverables. Minimum requirements for assurance case’s structure and contents are: to describe claims of IT products and systems properties, systematic argumentations of the claims, evidence and explicit assumptions of the argumentations; to structurally associate evidence and assumptions with the highest-level claims by introducing supplementary claims in the middle of a discussion. One common notation is Goal Structuring Notation (GSN) [4], which widely used in Europe for about ten years to verify system risk and validity after identifying risk requirements. Contents of GSN is shown below.

### Table 1. Contents of GSN

<table>
<thead>
<tr>
<th>Contents</th>
<th>Figure</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goal (Claim)</td>
<td></td>
<td>To describe claims of system and product properties</td>
</tr>
<tr>
<td>Strategy (Argumentation)</td>
<td></td>
<td>Systematic argumentations of the claims</td>
</tr>
<tr>
<td>Context (assumption)</td>
<td></td>
<td>Explicit assumptions of the argumentations</td>
</tr>
<tr>
<td>Undefined</td>
<td></td>
<td>Undefined claims and explanation</td>
</tr>
<tr>
<td>Evidence</td>
<td></td>
<td>Evidence of the argumentations</td>
</tr>
</tbody>
</table>

Minimum requirements for assurance case’s structure and contents are: to describe claims of system and product properties, systematic argumentations of the claims, evidence and explicit assumptions of the argumentations; to structurally associate evidence and assumptions with the highest-level claims by introducing supplementary claims in the middle of a discussion.

### 2.2 Security Assurance Case

Goodenough, Lipson and others proposed a method to create Security Assurance case [5]. They described that the Common Criteria provides catalogs of standard Security Functional Requirements and Security assurance Requirements. They decomposed Security assurance case by focusing on the process, such as requirements, design, coding, and operation. The approach did not use the Security Target structure of the CC to describe Security assurance case.

Alexander, Hawkins and Kelly overviewed the state of the art on the Security Assurance cases [6]. They showed the practical aspects and benefits to describe Security assurance case in relation to security target documents. However they did not provide any patterns to describe Security assurance case using CC. Kaneko, Yamamoto and Tanaka recently proposed a security countermeasure decision method using Assurance case and CC [7-8]. Their method is based on a goal oriented security requirements analysis [9]. Although the method showed a way to describe security assurance case, it did not provide Security assurance case graphical notations and the seamless relationship between security structure and security functional requirements.

### 2.3 Common Criteria

Common Criteria (CC: equivalent to ISO/IEC15408) [10] specifies a framework for evaluating reliability of the security assurance level defined by a system developer. In Japan, the Japan Information Technology Security Evaluation and Certification Scheme (JISEC) is implemented to evaluate and authenticate IT products (software and hardware) and information systems. In addition, based on CC Recognition Arrangement (CCRA), which recognizes certifications granted by other countries’ evaluation and authorization schemes, CC accredited IT products are recognized and distributed internationally. As an international standard, CC is used to evaluate reliability of security requirements of functions built using IT components (including security functions). CC establishes a precise model of Target of Evaluation (TOE) and the operation environment. And based on the security concept and relationship of assets, threats, and objectives, CC defines ST (Security Target) as a framework for evaluating TOE’s Security Functional Requirement (SFR) and Security
Assurance Requirement (SAR). ST is a document that accurately and properly defines security functions implemented in the target system and prescribes targets of security assurance. ST is required for security evaluation and shows levels of adequacy in TOE’s security functions and security assurance.

2.4 Risk Management

Risk Management's goal is to increase the impact and probability of positive risks and decrease them for negative risks. The point is not only avoiding failure, but to bring about opportunities. Time and energy can be spent avoiding, transferring to a third party, and mitigating potential failures. They can be similarly spent on accepting, sharing with third parties and enhancing opportunities. It is task of Risk Management to determine how much time and energy should be on avoiding failures and promoting opportunities.

Risk management includes six main processes in the theory of PMBOK [11]. These are risk management planning, risk identification, qualitative risk analysis, quantitative risk analysis, risk response planning, and risk monitoring and control.

3 ASSURANCE CASE FOR CURRENT SYSTEM RISK

3.1 Significance of Assurance case

Assurance case has been applied to the safety field [12] mainly. The largest benefit which we will enjoy by using the assurance case can perform the agreement argument for the demand between stakeholders enough. It is the point that can record the process of the reason / argument that reached the agreement result / conclusion. By certainly describing the following four points, assurance case offers the framework for building an argument more deeply.

- Claim
- Argumentation
- Evidence

- Explicit Assumption

We show significance of assurance case. Assurance case makes easy to confirm requirements by structured documentation. Requirements are verified that systems or services of target for evaluation are confirmed by assurance case. In addition, assurance case makes the basis clarified by the evidence that can achieve a goal. You can confirm the basis for judgment when a problem occurs. If the basis for judgment was validated by customer, you can use the evidence for legal basis.

The true difficulty of risk management in developing software comes from invisibility of software. Developers must build up software system avoiding many risks. Although software is invisible, developers must show that software system can work correctly as customer needs. It is important that customer’s need records correctly as an agreement. Assurance case record verified evidence with stakeholder. Therefore, assurance case is useful as a consensus building tool for risk management.

3.2 Current System Risks and Handling by Assurance Case

Risks in system development are categorized 3 types: Customer agreement risk, Business continuity risk, and System risk (Table2). Customer agreement risk has the risk of suits. Business continuity risk has the risk of communication. System risk has explanation risk of validation in developing activity.

However, in conventional development, only system risks are considered, the others are not considered. A brand is suffered a big economic loss when 3 type of risks are mixed. Therefore appropriate treatment for negative risks (avoid, transfer, mitigate) is important.

Assurance case for risk management is a method to avoid current risks in system development.
### 3.3 Difficulty of Treatment For Security Risks

It is especially important to handle security risks. If any security accidents happened, the brand would suffer a big economic and honor loss. It is also difficult to treat security risks.

Customer agreement risk of security of table 2 corresponds to suits risk when security accidents occur. To treat with this risk, it is necessary to have the evidences to show authentication of contractual customer’s agreements. Business continuity risk of security mainly is caused by service damages by attackers. To counter with this risk, it is necessary to have the evidences to show authentication of risk monitoring and control. System risk of security is the risk which occurs within development activities of IT products or security function of systems. To counter with this risk, it is necessary to have the evidences to show authentication of risk identification, analysis and counter plan. In this paper, the method to handle business continuity risk and system risk of security are focused.

It is important to show objective evidences that the customer recognizes that his request such as “The system is acceptably secure” is satisfied.

## 4 CC-CASE

### 4.1 Concept of CC-Case

We propose a methodology of security analysis and assurance, named CC-Case, using assurance case (ISO/IEC15026) and CC (Common Criteria: ISO/IEC15408).

Purpose of CC-Case is to solve several problems which we face at the development of secure systems which can handle more increasingly sophisticated threats. CC-Case can provide not only security requirement analysis method but also assurance according to the standard of CC. CC-Case contains the process which can clarify the scope of assurance for threats. It also contains the process which can verify the security specification based on CC using assurance case, and obtain an agreement of customer on the assurance.

The procedures of CC-Case have dual-layer. Upper layer is named logical model. Under layer is named concrete model. Logical model and concrete model is shown Figure 1. Logical model shows the process structure developed in detail as much as possible independently of specific system. Logical model has life-cycle process and each stage’s process. Concrete model contains real cases corresponding to the specific system. The concrete model is decomposed logically until it describes evidences at the bottom layer. It makes up evidences as real case and approval results of customers. These evidences recorded in sequence can be used for verification. Customer’s requirements may change frequently. It is necessary to keep evidences depending on changes. CC-Case supports changes through storing all evidences in a database.

Targets of CC-Case are IT products or systems. Although CC-Case is a method to make agreements with customers and developers, if we have no fixed customers, it is replaced by the person concerned on deciding requirements.
4.2 Life-Cycle Support of CC-Case

We show the support of the life-cycle process of CC-Case. The life-cycle process of CC-Case contains whole processes of requirement, design, implementation, test, maintenance stages. However, in this research, we focus on the process of requirement stage.

The life-cycle process of CC-Case should handle whole risks of security including business continuity risk of security.

Figure 3 is the life-cycle process of CC-Case. CC-Case uses GSN which is one common notation of assurance case. Using this assurance case, we explain the concept of the life-cycle process of CC-Case. In this case, top goal of assurance case is “IT products and systems using CC-Case are secure.” Explicit assumption of the arguments is “CC”. The strategy shows to verify the process of life-cycle development.

Figure 1. Logical model and concrete model

Figure 3. The life-cycle process of CC-Case
The strategy can be divided into 4 processes of the second goal, “Requirements using CC-Case is secure.”, “Designs using CC-Case is secure.”, “Implementations using CC-Case is secure.”, and “Tests and deliveries using CC-Case is secure”. These goals need evidence which can verify the goals.

The second goal of “Requirements using CC-Case is secure.” can be divided into 2 processes of the third goal, “Security specification using CC-Case is secure.”, and “Definition of development environment using CC-Case is secure.” through the strategy of “Verify security functions of requirements.” The third goal, “Security specification using CC-Case is secure.” is equivalent to the top goal of CC-Case at the requirement stage.

4.3 The Requirement Stage of CC-Case

(1) Assurance case of Security Specification

In the requirement stage, the procedures to make security specification are defined, and the documents which are necessary for ST (Security Target) are made. These procedures are defined as an assurance case, and produce evidences which give grounds of conformity with CC and agreements with customers. The assurance case of security specification can be classified into the stage of defining security concept, of making measures and of making security specification. Each stage’s logical model is shown. Also purpose, player, confirmation method for the output, and process (input, procedure, output) are clarified for every goal at the bottom layer of each stage.

Figure 4 shows the relationship of the procedures to make secure specification, inputs, evidences which give grounds.

Figure 4. Whole model of the requirement stage of CC-Case
Figure 5. Assurance case of Security Specification

- Security specification using CC-Case is secure.
- Verify the Security specification based on ST.
- Defining security concept is secure.
- Making security measures is secure.
- Making security specification is secure.

Figure 6. Defining security concept stage

- Outlines of Systems or Products/Needs of Customers/Trends of Market/Security requirements
- Defining security concept is secure.
- Verify the clarifying of security requirements.
- Target of evaluations is clear by setting the scope of analysis.
- Security requirements or issues is clear.
- Defining security concept is valid.
Figure 7. Stage of making security countermeasures

Figure 8. Stage of security specification
Figure 5 is the top assurance case of the requirement stage of CC-Case. By branched their sub goals, more detailed tasks are decided. Therefore the leftmost sub goal in Figure 5 is the top goal of Figure 6. The center sub goal in Figure 5 is the top goal of Figure 7. The light sub goal in Figure 5 is the top goal of Figure 8. It is the logical model that is the procedures of defined tasks in Figure 5, Figure 6, Figure 7, and Figure 8. It goes into details Figure 4.

The concrete model shows the evidences of ST contents and the logical relationship in the real cases into the ST contents under the all bottom goals. The evidences in Figure 4 are equivalent to the evidences of ST contents of the concrete model. The applied case of the concrete model is shown in Figure 9. The method of assurance case is applied to these procedures. Therefore we insist that the secure specification can be made by keeping the procedure. Upper goals need to satisfy all lower goals. At the same time, these goals are sufficient for all the requirements to do. These are the unique points which are different from previous methods like simple flow charts of processes. They show this method`s integrity. The more detailed processes of 3 sub goals in Figure 5 are shown as below.

(2) Defining security concept stage

The assurance case of the defining security concept stage is shown Figure 6. In this stage security requirements are extracts.

Taking into account of needs of customers, trends of market, security requirements, and security concept is defined.

In this stage the process makes not ST, but the security concept. The security concept is examined repeatedly many times and decided between customers and developers. In this stage, security requirements for products by the viewpoints of users are collected and arranged. After enough extractions, security requirements, the goal of “target of evaluations is clear by setting the scope of analysis “, and the goal of “Security requirements or issues is clear.“ are verified.

Then defining security concept is validated with agreement of customers. CC-Case considers validity confirmation as assurance.

(3) The stage of making security countermeasures

In this stage (Figure 7), with security concept as an assumption, CC-Case defines evaluation criteria, analyzes threats. It drafts, evaluates, and selects countermeasures. Then all the processes are verified that they are secure. There are 3 steps in this stage. In each step, the analysis and rationale of relationships between security requirements and relationships of logic are shown to customers. Each step defines the goal of validation that obtains an agreement to customers.

Step1: At first the scope of the target of evaluation (TOE) is defined. EAL is selected as the level of assurance. The definiteness of evaluation criteria is confirmed. It is necessary to get approval of a customer.

Step2: Define the threats models of information asset which is an object of protection and to perform analysis of a threat to the asset.

Step3: Select countermeasures to carry out from countermeasures plan. Countermeasures which do not handle are managed as remaining risks.

Next, the proof that proper selections are made is confirmed. It is necessary to get approval of a customer for the selections of security measures.

These steps of making countermeasures define the systematic procedure,

*1 Evaluation of criteria is validated.

*2 Evaluation of countermeasures is validated.

*3 Selection of countermeasures is validated.

These procedures intend that security measures are agreed upon between customers and evidences are left.

There are seven argument patterns of resolution and applied pattern [12] that Bloomfield showed on making assurance case. For the stage of making security countermeasures mentioned above, we made the comparison...
pattern of substitute plan that was one of the applied patterns [12] as a reference model. The stage of making security countermeasures is equivalent to a process to make ST after having examined threat analysis and evaluation criteria.

CC-Case did not just use ST item as a merely constituted process, we consider the following 5 points.

*1 Application of the type pattern of assurance case for securing of validity of the evaluation to measures.
*2 Validity confirmation process is set for the correspondence to a customer agreement risk by 3 steps.
*3 Making ST items for guarantees of the CC conformity
*4 Selecting enforcement measures
*5 Taking into account of remaining risks

(4) TOE summary specifications stage
In this stage (Figure8), extended component definition, security function requirement, security assurance requirement, summary specifications are verified to be secure. It is necessary to get customer agreement.

The security function requirement is made by selection of function requirement in CC part2 in order to establish the security objectives for the TOE as technical countermeasures.

The Security Assurance Requirement is made by references in CC part3.

When it is difficult to make the security function requirement and the Security Assurance Requirement using only CC, extended component is defined.

Then use extended function requirement and assurance requirement. Summary specifications show the method to implement the security function in the real system. Customer agreement is validated for this security specification.

4.4 Logical model detailed
The bottom goals of Figure 6-8 correspond to the top goals of the concrete model. Therefore the bottom goals are important, we examine those meanings closely.

We define purpose, player, confirmation method for the output, input, procedure, and output for the bottom goal. For example, we show the process on the bottom goal of “Threat analysis is secure.” which is the 5rd sub goal from the left in the stage of making security countermeasures.

“Threat analysis is secure.”

**Purpose:** We define a characteristic that TOE is going to deal with formal description technique, and the range of security.

**Player:** developer

**Confirmation method for the output:** verification

**Input:** assets, security function

**Procedure:** A threat consists of an adverse action performed by a threat agent on an asset. Therefore developer analyze a threat with an asset and a security function, extract an expected threat. Threat agents may be described as individual entities, but in some cases it may be better to describe them as types of entities, groups of entities etc. Examples of threat agents are hackers, users, computer processes, and accidents a threat is expressed by the name with prefix "T.".

**Example:** T.ACCESS An unauthorized user carries out access and the operation to resources

Output: a result of threat analysis

5 CONSIDERATION

5.1 Example of Concrete Model

Figure9 and Figure10 show an example of concrete model. CC-Case describes real case by the concrete model. As an example of real case, we show ST of IPA [13] using CC-Case. As a result, we confirm that we can write the whole example in assurance case.
Though a security case has a commonality, a process does not become clear. Therefore, there is a problem of the utility to have low unevenness and efficiency of the quality in individual treatment with the making security case.

To the contrary, CC-Case clarifies a commonality as a logical model, and there is the utility that can make a security case naturally by adding a concrete model. Also it has the good point to be able to leave the result as evidence.

Figure 9 is the concrete model that made the example that "the threat analysis is secure." in the stage of making security countermeasures of Figure 7. In this case, User data, TSF data, backup data are equivalent to the assets need to be protected. This verifies that the proper extraction of the threats are made and described. In this case the threats extracted are illegal logon, unauthorized access, misuse, injustice, spoofing, disclosing network data, removable medium, and unexpected accident.

Knowledge assets are made by obtained know-how using CC-Case. The extraction of threats such as the illegal logon becomes easy when we use the pattern of the threat that became a catalogue by knowledge assets. Each verification result is shown as evidence of the threats analysis such as T.ILLEGAL_LOGON, UNAUTHORIZED_ACCESS.

Figure 9 is equivalent to the example of the attached document of A.6.2 “threat” in CC Part 1. It includes verification means of the evidence in conformity with specifications of the ST.

The contents of the evidence of “T. ILLEGAL_LOGON” in the leftmost of Figure 9 is described as “An attacker may destruct, manipulate, and disclose a user data by pretending to be a fair user of TOE.”.

Figure 10 is the concrete model that made the example that "The Security Assurance Requirement is secure." in the stage of security specification of Figure 8. Figure 11 shows the contents of the evidence that is”6.2 Security assurance requirements”.

![Diagram](image.png)

Figure 9. Example of threat analysis
6.2. Security Assurance Requirements

TOE Security assurance requirements of TOE are shown as below. This TOE's Evaluation Assurance Level is EAL3. All security assurance requirements use the security assurance component defined CC part3 directly.

1. Development (ADV)
   - ADV_ARC.1: security architecture description
   - ADV_FSP.3: function specification with complete summary
   - ADV_TDS.2: architecture design

2. Guidance document (AGD)
   - AGD_OPE.1: user operation guidance
   - AGD_PRE.1: preparation procedure

3. Life-cycle support (ALC)
   - ALC_CMC.3: management of permission
   - ALC_CMS.3: CM Scope of implementation representation
   - ALC_DEL.1: delivery procedure
   - ALC_DVS.1: identification of security method
   - ALC_LCD.1: definition of life-cycle model by developer

4. Security Target Evaluation (ASE)
   - ASE_CCL.1: conformance claim
   - ASE_ECD.1: extended component definition
   - ASE_INT.1: ST introduction
   - ASE_OBJ.2: security objectives
   - ASE_REQ.2: derived security requirements
   - ASE_SPD.1: security problem definition
   - ASE_TSS.1: TOE summary specifications

5. Test (ATE)
   - ATE_COV.2: coverage analysis
   - ATE_DPT.1: test: base design
   - ATE_FUN.1: function test
   - ATE_IND.2: independent test – sample

6. Vulnerability appraisal (AVA)
   - AVA_VAN.2: vulnerability analysis

Figure10. Example of security assurance requirements

Figure11. Evidence of security assurance requirement
5.2 The Merits of CC-Case

CC-Case has many merits to solve several problems which we face at the development of secure systems. In this paper, we show the merits focus on its life-cycle process.

The life-cycle process of CC-Case can be expected to establish the discipline and control in the processes of refinement of the IT products and systems during its development and maintenance. It strengthens the handling for system risk and business continuity risk by life-cycle support.

The merits of handling system risk are mentioned below. CC-Case makes possible to improve the development method to handle. Life-cycle process using CC-Case establishes rules and control in the development and assurance of objective system and production by implementing its requirements to system and production correctly.

For example, at design stage CC-Case makes easier to accept specification changes by using its logical traceability and evidences. It can expect the improvement of development method, reuse, productivity that is the problem of the assurance case.

By defining development processes at each stage, CC-Case would be improved to a development method with assurance of life-cycle. It can keep assurance based on CC through the life-cycle process. If its scope was only requirement stage, it has only assurance at the time as the mere expectation. If its scope is whole life-cycle process, it is possible to assure the real products of long span. In other words, CC-Case with life-cycle support has different quality of assurance from CC-Case with requirement stage.

The merits of handling business continuity risk are mentioned below. Security risks changes incessantly because invisible attackers exist, and an unexpected new threat occurs. The life-cycle support of requirement, design, test, and maintenance makes countermeasure easily against the situation which an unexpected new threat produced.

Life-cycle process of CC-Case can handle the business continuity risk by its monitoring and control process. It makes easier to handle the change of risks additionally.

If you make a system or a production with CC-Case, it makes easy to cope with modification for a security accident. Because CC-Case has evidences of the argumentations to clarify a modification point for an essential cause to a security accident. At maintenance stage, CC-Case can be expected to improve reusability, productivity by reusing evidences stored according to defined process.

6 CONCLUSION

6.1 Summary

We propose a methodology of security analysis and assurance, named CC-Case, using assurance case (ISO/IEC15026) and CC (Common Criteria: ISO/IEC15408).

We explained the general concept. CC-Case has dual-layer. Upper layer is named logical model. Under layer is named concrete model. Logical model shows the process structure developed in detail as much as possible independently of specific system. Logical model has life-cycle process and each stage’s process. Concrete model contains real cases corresponding to the specific system. It makes up evidences as real case and approval results of customers. We show the detailed explanation of logical model, example of concrete model, and merits of the life-cycle process of the CC-Case.

CC-Case strengthens the handling for system risk and business continuity risk by life-cycle support.

6.2 Future Tasks

There are some unsolved issues in the CC-Case presented in this paper.
(1) We need to show detailed processes of stages over life-cycle except requirement.
(2) A detailed selection process for remaining risks is important as a future task, because it implies specific assurance. In addition, describing measures in detail applied when unknown threats actually occur is important as a future task. We need to define remaining threats and reaction procedures for events caused by the remaining risks.

7 REFERENCES

2. OMG, ARM, http://www.omg.org/spec/ARM/1.0/Beta1/