International Journal of Systems and Software Security and Protection
Volume 12 • Issue 1 • January-June 2021

CSPM:
Metamodel for Handling Security and Privacy Knowledge in Cloud Service Development

Tian Xia, Waseda University, Tokyo, Japan
Hironori Washizaki, National Institute of Informatics, System Information, eXmotion, Waseda University, Tokyo, Japan
Yoshiaki Fukazawa, Waseda University, Tokyo, Japan
Haruhiko Kaiya, Kanagawa University, Yokohama, Japan
Shinpei Ogata, Shinshu University, Matsumoto, Japan
Eduardo B. Fernandez, Florida Atlantic University, USA
Takehisa Kato, Hitachi, Ltd., Tokyo, Japan
Hideyuki Kanuka, Hitachi, Ltd., Tokyo, Japan
Takao Okubo, Institute of Information Security, Yokohama, Japan
Nobukazu Yoshioka, National Institute of Informatics, Tokyo, Japan
Atsuo Hazeyama, Tokyo Gakugei University, Koganei, Japan

ABSTRACT

Security and privacy in cloud systems are critical. To address security and privacy concerns, many security patterns, privacy patterns, and non-pattern-based knowledge have been reported. However, knowing which pattern or combination of patterns to use in a specific scenario is challenging due to the sheer volume of options and the layered cloud stack. To deal with security and privacy in cloud services, this study proposes the cloud security and privacy metamodel (CSPM). CSPM uses a consistent approach to classify and handle existing security and privacy patterns. In addition, CSPM is used to develop a security and privacy awareness process to develop cloud systems. The effectiveness and practicality of CSPM is demonstrated via several case studies.

KEYWORDS

Cloud Computing, Privacy Patterns, Security Patterns, Software and System Architecture, Software Patterns

1. INTRODUCTION

Cloud service providers control remotely available services and data, which are often connected with other services. Consequently, ensuring security and privacy (S&P) in cloud services is critical. Many of the cloud security and privacy issues are also true for any kind of distributed system; however, cloud architectures bring new attacks (Fernandez, Monge & Hashizume, 2016). Besides, clouds may store large amounts of sensitive information such as users' personal information. Thus, the result of

DOI: 10.4018/IJSSSP.20210101.oa1

a successful attack could be catastrophic because an attacker may compromise data from many users (Fernandez, Monge & Hashizume, 2016).

Because software engineers are not necessarily experts in S&P, resolving S&P concerns throughout the software lifecycle is challenging. Software patterns are abstractions from recurring concrete problems and corresponding solutions that appear in non-arbitrary contexts (Riehle & Zullighoven, 1996) (Fernandez, Yoshioka & Washizaki, 2008) (Nhlabatsi, et al., 2010) (Fernandez, et al., 2014) (Fernandez, et al., 2018) (Washizaki, 2017) (Washizaki, et al., 2018). Besides the numerous cloud S&P patterns reported to date (Hashizume, Yoshioka & Fernandez, 2011) (Hashizume, Yoshioka & Fernandez, 2012) (Reimer, Abraham & Tan, 2013) (Fernandez, Yoshioka & Washizaki, 2014) (Fernandez, Yoshioka & Washizaki, 2015) (Fernandez, Yoshioka & Washizaki, 2015) (Fernandez, Yoshioka & Washizaki, 2016) (Rath, 2018), non-pattern-based knowledge (e.g., practice and principles) is used to handle S&P issues in cloud service development. The sheer volume of S&P patterns and non-pattern-based knowledge makes selecting the appropriate knowledge or combination of patterns and knowledge challenging. Although this issue is relevant to S&P patterns in general, it is more critical in cloud services. First, cloud services and their underlying mechanisms are integrated over multiple layers in a layered cloud stack. Second, a cloud system links numerous devices, and each device has its own deployment model and service. This intertwined system leads to many concerns, including S&P.

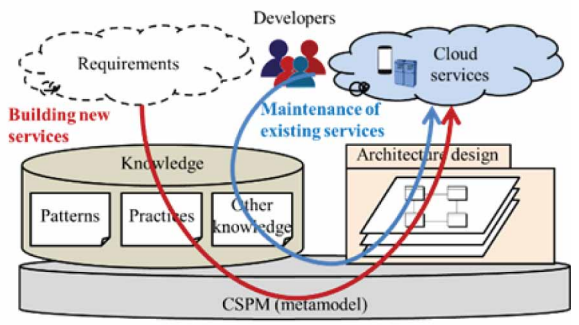
The above issues can be mitigated via reference architectures or metamodels that capture and encapsulate the essential concepts related to S&P in layered cloud stacks. Previously, we reported an earlier version of the metamodel (Washizaki, et al., 2016) (Xia, et al., 2018). This study proposes an extension called the "Cloud Security and Privacy Metamodel (CSPM)" to address S&P in cloud services. CSPM integrates and extends existing cloud security metamodels with newly added concepts. CSPM can be used for supporting cloud service development and maintenance (Figure 1). CSPM describes S&P-related knowledge over multiple layers. Besides selecting and combining the appropriate patterns to address S&P issues, CSPM can be used for designing high-level architectures of cloud service systems effectively and efficiently.

As an extension to our previous research, we conducted experiments and a case study to address the following questions:

RQ1: Can CSPM resolve S&P problems and help application of the corresponding patterns?
RQ2: Can CSPM improve the system by efficiently providing S&P solutions?
RQ3: Can CSPM and the corresponding process using CSPM be deployed in practical real-world applications?

RQs 1 and 2 evaluate CSPM from two viewpoints. RQ3 demonstrates the usability of our approach for the metamodel itself and the process we propose. The case study, which involves an application

Figure 1. Overview of cloud services and our metamodel



similar to a commercial one using a conventional cloud platform, suggests that CSPM has practical applications in industrial development. Tools such as this metamodel should contribute to the ubiquity of patterns to develop secure systems.

The novel contributions of this paper are as follows:

1. We proposed CSPM, which is a metamodel as the basis for describing S&P-related knowledge over multiple cloud layers. To the best of our knowledge, CSPM is the first metamodel to uniformly handle security-related concepts as well as privacy-related ones over multiple layers.
2. We proposed a S&P awareness process by using CSPM for developing cloud services.
3. We conducted a controlled experiment and a case study based on the proposed process to evaluate the effectiveness of the problem analysis and solution design supported by CSPM.

The rest of this paper is organized as follows. Section 2 contains related work and problems addressed in this research. Section 3 proposes our metamodel and overviews our process for S&P development. Section 4 discusses our case studies and answers our RQs, and section 5 concludes this paper.

2. RELATED WORK AND CHALLENGE

2.1. Related Work

Several metamodels and conceptual models have addressed both S&P (Kalloniatis, Kavakli & Gritzalis, 2008) (Tesoriero, 2011) (Islam, et al., 2018). However, they are difficult to apply directly to cloud services.

Cloud security is considered in several metamodels and abstract reference architectures (Hazeyama, 2012) (Chatziprimou, Lano & Zschaler, 2013) (Fernandez, Monge & Hashizume, 2016). However, cloud privacy along with security has yet to be considered. Due to their intertwined relationship, they should be addressed simultaneously.

One study surveyed software security knowledge and proposed a metamodel to model such knowledge (Hazeyama, 2012). Unlike that study, which did not include computing, our study incorporates such knowledge into our metamodel. Another study used a metamodel to model cloud services and resources (Chatziprimou, Lano & Zschaler, 2013), but neither security nor privacy were considered directly. A different study reported an abstract security reference architecture model to develop secure cloud services and systems (Fernandez, Monge & Hashizume, 2016). This study provided a basis to model multiple layers of the cloud in terms of the security at each layer. However, privacy was not addressed.

There are several modeling frameworks for cloud security that auditing mechanisms (Ismail & Islam, 2020) (Mouratidis, Shei & Delaney, 2020). Although these frameworks identify key security-related concepts, privacy-related concepts and the layered cloud stack were not addressed explicitly.

Some studies have focused on privacy engineering. One did a systematic literature mapping on privacy patterns research (Lenhard, Fritsch & Herold, 2017). However, this study did not consider a metamodel or security patterns. Another study proposed a metamodel for privacy engineering based on SEMDM, which is a metamodel for software and systems development methodologies (Martín & del Álamo, 2017). This study did not consider privacy patterns, security patterns, or cloud computing. A different study proposed a privacy engineering metamodel by extending SEMDM (Alamo, Martín & Caiza, 2017). Although it included privacy design strategies, privacy threats, and privacy design patterns as well as listed elements similar to our metamodel, relationships were not considered.

A study proposed a metamodel for General Data Protection Regulation (GDPR)-based privacy level agreements (PLAs) to support privacy management, based on analysis of privacy threats, vulnerabilities, and trust relationships in general information Systems (Diamantopoulou,

Angelopoulos, Pavlidis, & Mouratidis, 2017). This study does not address patterns or cloud-specific concerns. By connecting our metamodel with the proposed one, we can consider incorporating GDPR-based PLAs in cloud service development.

2.2. Challenge

Often a developer who is inexperienced and not an expert in S&P is tasked to build a cloud application. As the developer is aware of her shortcomings, she searches for such documents on S&P. However, this leads to several problems:

- **Numerous S&P Patterns and Documents:** Patterns are reusable solutions to recurring problems. Because many S&P patterns (and other documentation) have been proposed, the search results are overwhelming. Selecting the appropriate pattern(s) when many are not applicable to cloud services (Fernandez, et al., 2010) (Fernández, et al., 2016) is difficult, especially for a novice developer.
- **Complex Relationships Between a Cloud Service and its Mechanism:** A cloud is composed of three main layers: infrastructure, platform, and software. Although each service is provided from one layer from the users' viewpoint, a service may control data related to other layers (Subashini & Kavitha, 2011) (Fernández, Yoshioka & Washizaki, 2019). Consequently, selecting and utilizing the appropriate pattern(s) are challenging tasks.
- **Practical Metamodels for Cloud Development do not Exist:** Existing metamodels (Kalloniatis, Kavakli & Gritzalis, 2008) consist of essential concepts when dealing with S&P issues. However, they cannot deal with real-world S&P issues in cloud development.

3. CLOUD SECURITY AND PRIVACY METAMODEL (CSPM) AND PROCESS

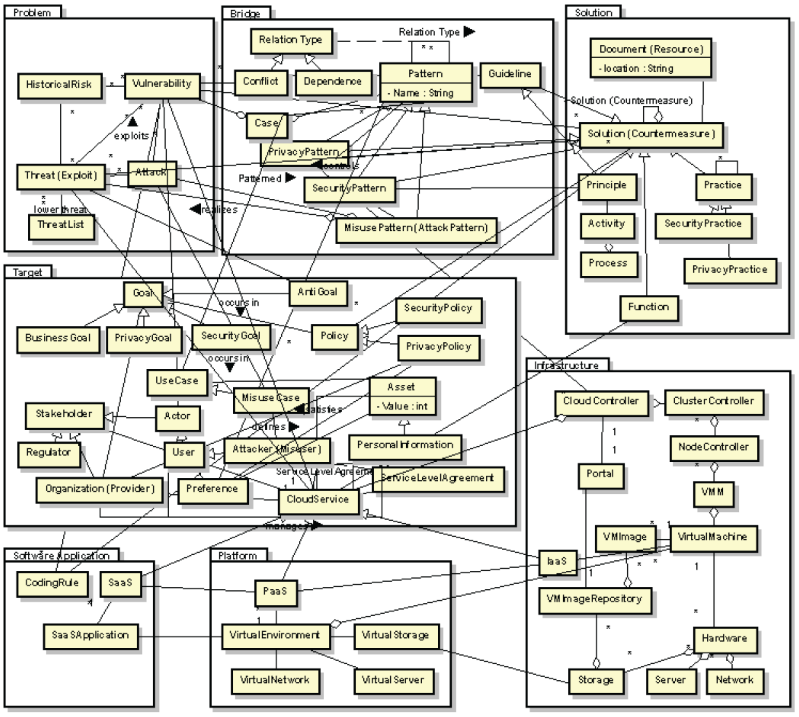
3.1. Design of the Metamodel

CSPM is a metamodel as the basis for describing S&P-related knowledge over multiple layers. Besides selecting and combining the appropriate patterns to address S&P issues, CSPM can be used to design architectures of cloud service systems effectively and efficiently. Figure 2 shows the overview of Cloud Security and Privacy Metamodel (CSPM) as a set of seven packages in the form of a UML class diagram. Table 1 outlines these packages by showing major concepts in them.

Table 1. Packages in the metamodel

Package	Outline	Major Concepts
Problem	Common concepts for problems	Threat, vulnerability, attack
Bridge	Concepts on the relationships between problems and corresponding solutions	Pattern, case, guideline
Solution	Common concepts for solutions	Solution (countermeasure), security function, practice
Software Application	Concepts specific to the software application layer providing on-demand applications	SaaS application, coding rule
Platform	Concepts specific to the platform layer offering virtual environments	Virtual environment, virtual storage, virtual storage
Infrastructure	Concepts specific to the infrastructure layer providing virtualized resources that can be assigned to virtual machines (VM)	Virtual machine, hardware, storage
Target	Concepts specific to the target application	Goal, policy, asset, cloud service

Figure 2. Overview of Cloud Security and Privacy Metamodel (CSPM)

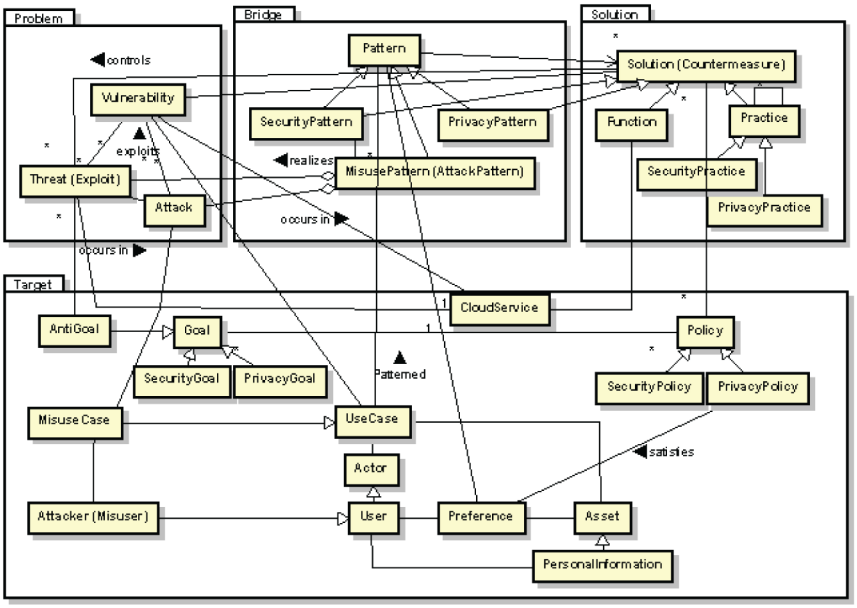


Also, Figure 3 shows a simplified version of CSPM named Privacy View, which is a simplified metamodel that emphasizes privacy-related concepts (such as personal information) and their surrounding elements. As shown in Figure 3, the privacy-related concepts are related to the security-related concepts in CSPM. For example, an attacker may access personal information against its users' preferences via a misuse case. Such privacy threats can be mitigated by applying appropriate corresponding security patterns.

CSPM addresses the aforementioned challenge by having the following features:

- **Consistency Over Multiple Layers:** The problem, bridge, and solution packages are fundamental in all layers. Not only do they provide concepts common between layers, but they also organize their relationships. Consequently, they uniformly handle S&P-related knowledge over different layers.
- **Convenience:** Separating general concepts from specific ones (e.g., layers, cloud-specific knowledge, and cloud-independent knowledge) into packages makes the metamodel easy to access.
- **Compatibility With Existing Cloud Services and Security Metamodels:** In addition to consistency, the packages include concepts according to the relationships deðned in existing reference architecture and metamodel (Fernandez, Monge & Hashizume, 2016) (Hazeyama, 2012). Hence, the proposed metamodel can work with existing metamodels. For example, the platform package and the infrastructure package of CSPM encapsulate concepts that are identified as PaaS-related and IaaS related respectively in the existing reference architecture (Fernandez, Monge & Hashizume, 2016).

Figure 3. Overview of the Privacy View model



3.2. Modeling Based on the Metamodel

CSPM can be a basis for modeling vulnerabilities from databases such as the Common Vulnerabilities and Exposures (CVE) (MITRE, 1999). For example, a vulnerability Cross-site Scripting (XSS) (MITRE, 2012) can be modeled in Figure 4. In the figure, elements related to the vulnerability are modeled with stereotypes specifying corresponding concepts in CSPM. To identify problems and implement countermeasures easily, the model in the figure helps visualization of vulnerable elements.

In addition, CSPM can help users to depict the pattern problem and solution (Figure 5). In the figure, elements related to the Authenticator pattern (Schumacher, et al., 2006) are modeled with stereotypes specifying corresponding concepts in CSPM.

Figure 4. Model of a Cross-site scripting (XSS) vulnerability based on CSPM

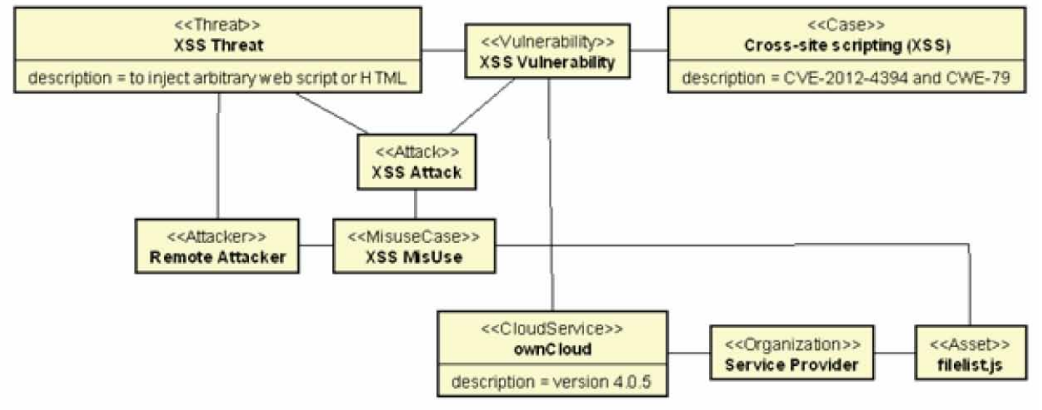
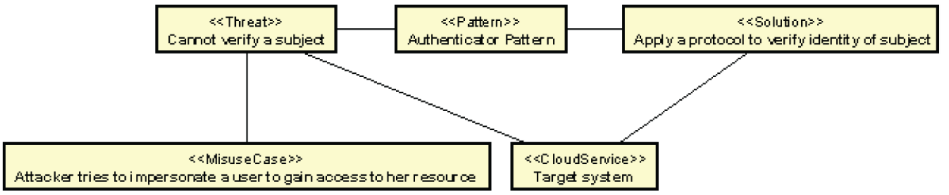


Figure 5. Model of an Authenticator pattern based on CSPM



3.3. S&P Development Process

We propose a S&P awareness process by using CSPM for developing cloud services (Figure 6). S&P development consists of four phases: analysis, design, implementation, and testing. Each phase is described below:

1. S&P Requirement Analysis: While analyzing the system requirements, the threats and S&P problems in the current system model are identified using a threat model such as STRIDE (Microsoft, 2002) (Jelacic, et al., 2017) together with concepts related to vulnerabilities organized in CSPM.
2. S&P Design: S&P patterns and other knowledge descriptions can be used to determine possible solutions. Concepts related to S&P patterns organized in CSPM can help select appropriate S&P patterns corresponding to the identified threats and problems from the knowledge base.
3. S&P Implementation: The system is implemented according to the determined solutions.
4. S&P Testing: The system is tested. If problems arise during the test, return to the phase (1).

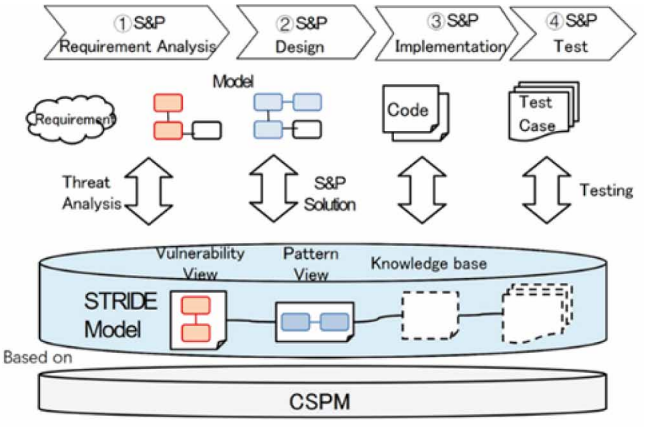
5. EXPERIMENT AND CASE STUDY

To evaluate the effectiveness of the problem analysis and solution design supported by CSPM, we conducted an experiment and a case study.

5.1. Experiment

A controlled experiment evaluated the impact of CSPM and investigated the RQs.

Figure 6. Overview of the S&P Development Process



5.1.1. Experiment Setting

The experiment was designed to evaluate the impact of CSPM. The experiment involved two groups of college students, ranging from fourth year undergraduate to second year master's students. The groups were labeled as the experiment group (EG) and the control group (CG).

Regardless of the group, participants were asked to read the class diagram and use case explanation to determine the S&P issues in the system model. The system model was simplified from student work and contained several security threats. The participants were asked to resolve S&P issues on the model level. As a reference, we prepared some S&P patterns, but not all were applicable to this system. After the experiment, participants completed a questionnaire.

EG received additional support. They were given CSPM, a simplified version of CSPM named Pattern View, and a guideline showing how to apply a pattern with an example. The Pattern View is a simplified metamodel that emphasizes elements related to S&P patterns such as goals, threats, and solutions (Figure 7). Because it can analyze the requirements and threats to a system, applicable S&P patterns can be determined. It can depict the pattern problem and solution.

5.1.2. Experiment Results

The results for CG are shown in Table 2 and Table 3, while those of EG are shown in Table 4 and Table 5. In Table 2 and Table 4, four variables (the total time to complete the assignment, number of problems identified in the system, number of problems solved by revising the model, and number of patterns used to solve problems) were measured. Also, Table 3 and Table 5 show what kind of problems were identified and solved by each participant in detail.

Figure 7. Overview of the pattern view model

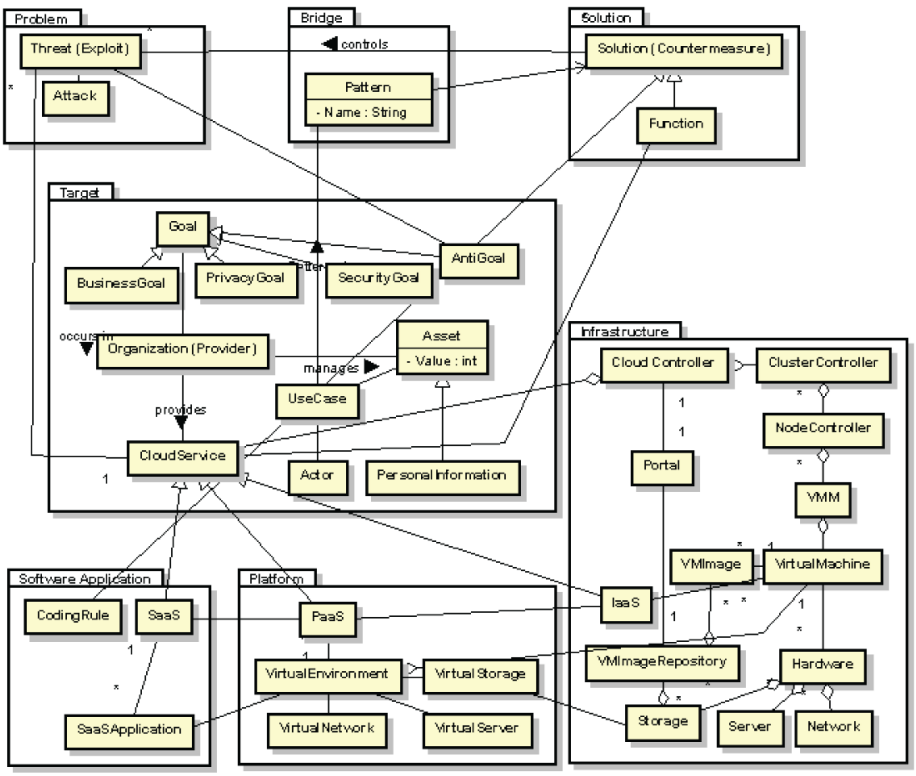


Table 2. Results for the control group (CG)

Participant	Time [min]	Problems Identified	Problems Solved	Pattern Used
C1	100	5	3	3
C2	180	2	1	0
C3	60	5	5	0
C4	60	3	2	0
C5	60	3	1	1
Average	92	3.6	2.5	0.8

Table 3. Problems identified and solved by the control group (Pattern: Participants solved by applying patterns, Solve: Participants solved without specific pattern applications, Identify: Participants identified problems but never solved, Fail: Participants failed to find problems)

	Problem Related to Authentication	Problem Related to Authorization and Access Control	Problem Related to Password	Problem Related to DDoS	Other Problems
C1	**Pattern**	**Pattern**	**Pattern**	Identify	Identify
C2	**Pattern**	Identify	Fail	Fail	Fail
C3	**Solve**	**Solve**	**Solve**	**Solve**	**Solve**
C4	**Solve**	**Solve**	Fail	Identify	Fail
C5	**Pattern**	Identify	Identify	Fail	Fail

Table 4. Results for the experiment group (EG)

Participant	Time [min]	Problems Found	Problems Solved	Pattern Used
E1	90	3	3	3
E2	70	3	3	2
E3	60	4	3	3
E4	60	4	3	3
E5	50	5	5	3
Average	66	3.8	3.4	2.8

Table 5. Problems identified and solved by the experiment group

	Problem Related to Authentication	Problem Related to Authorization and Access Control	Problem Related to Password	Problem Related to DDoS	Other Problems
E1	**Pattern**	**Pattern**	**Pattern**	Fail	Fail
E2	**Pattern**	**Solve**	**Pattern**	Fail	Fail
E3	**Pattern**	**Pattern**	**Pattern**	Identify	Fail
E4	**Pattern**	**Pattern**	**Pattern**	Fail	Identify
E5	**Pattern**	**Pattern**	**Pattern**	**Solve**	**Solve**

Some of the participants (C1 and C2) read all the reference patterns. C1 spent a long time on the assignment and used the patterns. However, C2 was confused about pattern use and did not use the reference patterns to complete the task. On the other hand, other participants (C3–C5) did not review the reference patterns. Due to previous development experience, C3 did not need the reference patterns to be successful. C4 and C5 finished quickly. Although they addressed the main S&P issues, they did not address minor problems.

The results of the EG group were similar. They solved a minimum number of principle problems with a greater emphasis on S&P patterns and revised the model correctly. Most completed the experiment in about an hour. Some issues not related to the reference patterns (e.g., DDoS attack) were not solved.

Although the difference between EG and CG to solve problems was not significantly different, EG was more proficient. Three or more main S&P issues were resolved by the EG participants, whereas the number of issues addressed fluctuated widely within the CG group. This difference is attributed to the S&P patterns.

Although we speculated that the EG group would complete the tasks faster than the CG group, the completion time between the two groups were statistically insignificant. This may be attributed to the time that the EG group spent reading the metamodel and guideline. Comparing C1, who used patterns for assignment, to the EG group indicates that applying our approach is less time consuming because C1 spent a lot of time reading the reference material.

All participants in the EG group provided similar responses to the questionnaire. All indicated that the Pattern View of the metamodel itself (Figure 7) is easy to understand, but it has low utility. On the other hand, the explanation and example in the guideline are very helpful, especially for applying patterns. Participants responded that the Pattern View structure of the S&P pattern is helpful, but it is preferable to use this in conjunction with a detailed description of the patterns.

5.2. Case Study: "Treasure-Hunting Game"

To evaluate the effectiveness of the problem analysis and solution design supported by CSPM, we conducted a case study for developing cloud service applications targeting an Android game that stores data in a cloud. The original unsecured version and that security enhanced by CSPM were used to evaluate.

To confirm the contribution of CSPM, a student work (the "Treasure-Hunting Game") was used. Similar to popular commercial games (e.g., Pokémon Go and Ingress), this game is an AR application where streets contain multiple spots, and one spot has the hidden treasure. The first author of this paper designed the initial structure and interface as shown in Figure 8. To begin, players input their names in order to manually save their data like hints and coins into the cloud and to check target player's data. In this case study, cloud functions were implemented on Amazon Web Service (AWS).

The STRIDE model was used in the S&P requirement analysis (Table 6). Because Android API and AWS API addressed the threats due to listening to transmissions or tampering with local data, the case study was concerned with the authentication problem and access right problem, as described below:

- **Authentication Problem:** Because anyone can use this game, the identity spoofing risk is high, which may lead to data tampering in cloud storage.
- **Solution by Pattern:** The Authenticator Pattern adds an authenticator to require a user to sign up and sign in before accessing the system. Other patterns like Password Design and Use may also provide support.
- **Access Right Problem:** The original game only requires a user name to display user data on the screen. This feature may be designed so that friends' data can be checked, but anyone can check a user's information.

Figure 8. Initial design of the Treasure Hunting Game software system

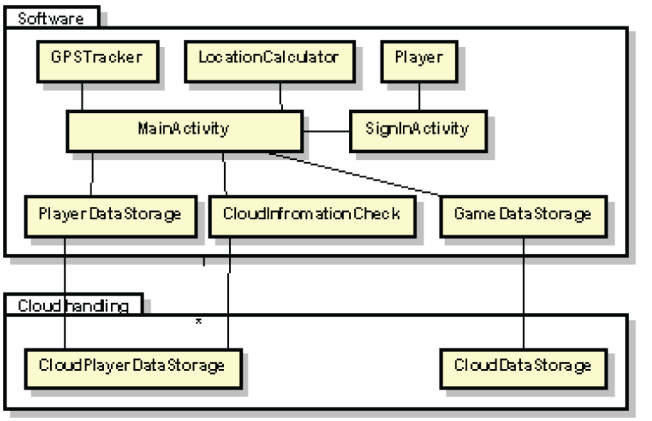


Table 6. Result for the S&P requirement analysis based on the STRIDE model and concepts in CSPM

Goal	Anti-Goal	Security Problem	Specific Example	Security Pattern	Solution
Tamper proof data	Gain ability to tamper with data	Unauthorized actors tampering with local data	User accesses local data on their phone, changing their score	Encryption pattern	Provided by the Android phone itself
		Unauthorized actors tampering with cloud data	User logs in as another user. Cloud user data might be modified.	Authentication pattern	Require a password for each user
Confidentiality	Gain access to confidential information	Unauthorized actors listening to the transmissions	Man in the middle attack	Transmission pattern	API automatically uses SSL and can be set to use a VPN
		Information disclosure	User accesses other players' data without permission	Authorization pattern, RBAC pattern	Control access rights for each player
		Elevation of privilege	User pretends to be an administrator and granted unlimited access to all game data	Authentication pattern, Transmission pattern	Player can only access the database which is limited by the permission levels of a third-party server
Reliability	Gain ability to access other player's data	Identity spoofing	Anyone can access the game	Authenticator pattern	Require sign up and sign in
Availability	Bring down the servers	Denial of service	Server becomes flooded by non-legitimate messages	Firewall, DDoS patterns	Unrealistic issue due to the game's small scale

- **Solution by Pattern:** The Authorization Pattern and Role-based Access Control (RBAC) Pattern (Schumacher, et al., 2006) (Yoshioka, Washizaki & Maruyama, 2008) can limit access rights.

Figure 9 shows use cases and misuse cases based on the requirement analysis. Figure 10 shows the results of our analysis of goals, problems, patterns, and solutions by referring to the STRIDE model as well as concepts in the Pattern View of CSPM. In Figure 10, we can confirm that how security problems imposed by misuse cases (e.g., "Use without permission") are characterized by concepts in the Problem package such as attacks and threats. We can also trace how these problems would be mitigated by solutions of specific security patterns (e.g., Authorization Pattern and RBAC Pattern). Then, the first author modified the design model to incorporate the identified solutions as shown in Figure 11. In these figures, elements related to the security patterns are modeled with stereotypes specifying corresponding concepts in the metamodel. We confirmed that the authentication problem and the access right problem are resolved, and the access controller works as intended.

In terms of multiple cloud layers, these solutions have been mainly achieved by the concepts in the platform package and supported by other underlying concepts in the software application and infrastructure packages. It shows how users of CSPM can handle S&P-related knowledge over different layers. Figure 10 and Figure 11 show that "SaaS" from the software application package and "Storage" from the infrastructure package have been utilized together with concepts in the platform package to address threats and attacks indirectly.

In terms of privacy, the player data is personal information while the Authorization Pattern is the applied security pattern to prevent players from accessing other players' data.

The ability of CSPM to revise the model was investigated via the case study. The proposed process was used in the case study, demonstrating that CSPM is applicable to S&P analysis and during cloud system development, respectively. The problems in the original target system are addressed in the revision. CSPM is effective, at least for a simplified system. However, not all the components of the cloud system were considered by CSPM in the case study. As a system becomes more complex, other issues may arise. Hence, the entire metamodel should be further evaluated in the future.

5.3. Discussion

RQ1: Can CSPM resolve S&P problems and help application of the corresponding patterns?

In the experiment, EG solved more problems than CG in the same or less time. EG participants selected and applied the appropriate pattern to revise the model due to the support of our approach.

Figure 9. Use cases and misuse cases of the Treasure-Hunting Game

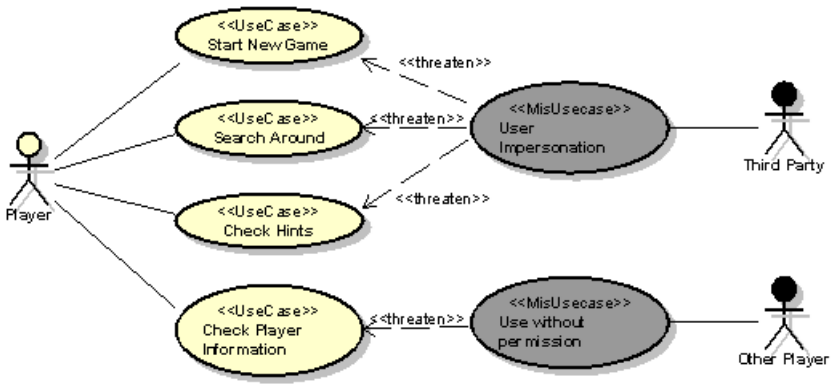
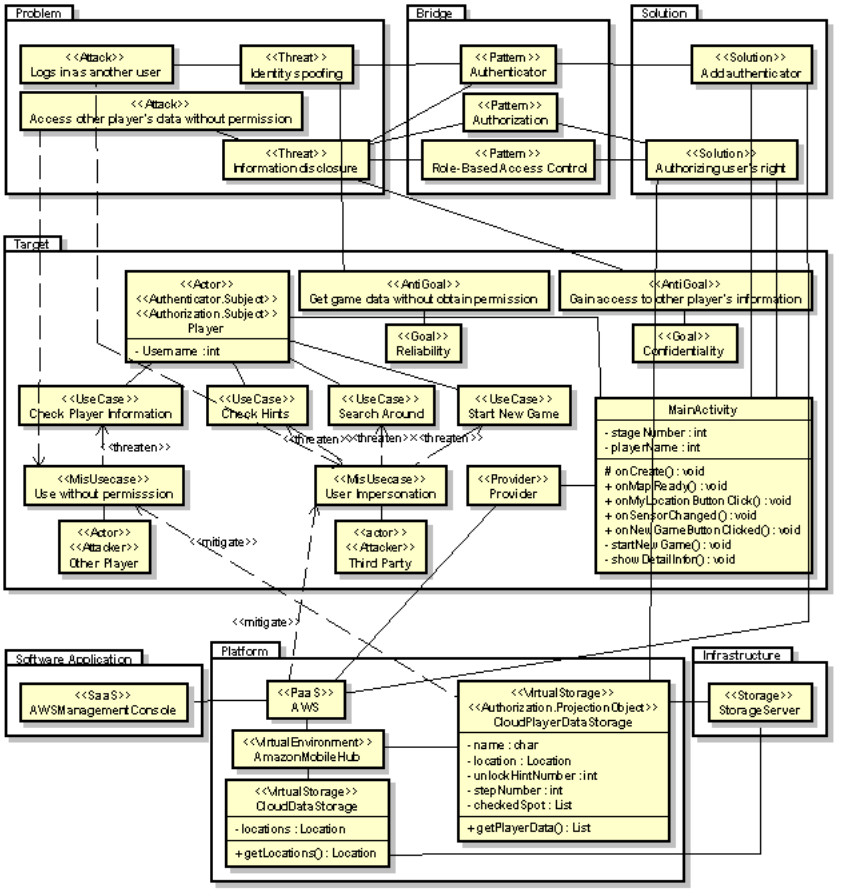


Figure 10. Results of analysis of goals, problems, patterns, and solutions based on CSPM



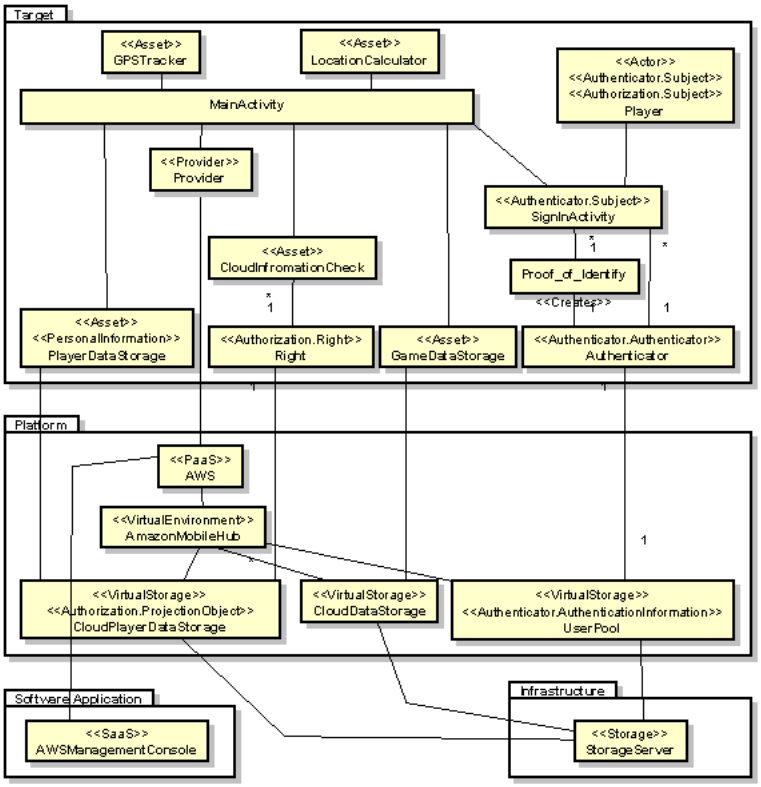
Although the knowledge base in this study is small, the proposed method should provide improved results when dealing with more S&P patterns. Our approach, especially the Pattern View structure of the S&P pattern, can identify necessary patterns and improve pattern comprehension.

Unlike previous research, which used metamodels for security issues, this study used CSPM to combine security (i.e., authentication) and privacy (i.e., access right control). This study only considered simple combinations of S&P patterns, which were indicated previously (e.g., *Authenticator Pattern* with *Single Access Point Pattern*), due to the small scale of the target system. In the future, more complex systems should be evaluated.

RQ2: Can CSPM improve the system by efficiently providing S&P solutions?

The ability of Pattern View of CSPM to revise the model was investigated via a case study. The problems in the original target system are addressed in the revision. CSPM is effective, at least for a simplified system. Not all the components of the cloud system were considered by CSPM in the case study. As a system becomes more complex, other issues may arise. Hence, the entire metamodel should be further evaluated in the future.

Figure 11. Modified secure design of the Treasure Hunting Game software system in detail



RQ3: Can CSPM and the corresponding processes using CSPM be deployed in practical real-world applications?

The proposed process was used in the experiment and case study, demonstrating that CSPM is applicable to S&P analysis and during cloud system development, respectively. Both indicate that CSPM is practical in some situations. However, the participants in the experiment provided negative feedback about the metamodel's usefulness. They felt that the current guideline is more useful than the metamodel. Revising the guideline to provide more examples of CSPM usage should improve the practicality of our approach.

6. CONCLUSION AND FUTURE WORK

CSPM, which deals with S&P in cloud services, can be used in software development. Its effectiveness and usability are confirmed via a case study and an experiment. The case study, which involves an application similar to a commercial one using a conventional cloud platform, suggests that CSPM has practical applications in industrial development.

There are several future directions. The first is to implement larger complex case studies such as a development of a cloud system with multiple services to evaluate the effectiveness of CSPM. The second is to apply concepts in CSPM semi-automatically to detect specific threats. The third is to develop a detailed framework to broaden the usage of CSPM.

ACKNOWLEDGMENT

The authors thank Dr. Masayuki Yoshino and Dr. Dan Yamamoto for their helps. They also would like to thank the anonymous reviewers for their insightful comments and suggestions. This research was supported by the SCAT Research Grant; the MEXT enPiT-Pro Smart SE: Smart Systems and Services innovative professional Education program; the JSPS KAKENHI [grant number 16H02804]; the JSPS KAKENHI [grant number 17K00475]; the JST-Mirai Program [grant number JP18077318]; and the JST-Mirai Program [grant number JP20319852].

REFERENCES

Alamo, J. M., Martín, Y. S., & Caiza, J. C. (2017). Towards Organizing the Growing Knowledge on Privacy Engineering. In *Proceedings of the IFIP International Summer School on Privacy and Identity Management*, (pp. 15-24). Springer.

Almutairi, A. A., Sarfraz, M., Basalamah, S., Aref, W., & Ghafoor, A. (2012). A Distributed Access Control Architecture for Cloud Computing. *IEEE Software*, *29*(2), 36–44. doi:10.1109/MS.2011.153

Chatziprimou, K., Lano, K., & Zschaler, S. (2013). Towards a Meta-model of the Cloud Computing Resource Landscape. *Proceedings of the 1st International Conference on Model-Driven Engineering and Software Development (MODELSWARD)*.

Diamantopoulou, V., Angelopoulos, Pavlidis, M., & Mouratidis, H. (2017). A Metamodel for GDPR-based Privacy Level Agreements. *Proceedings of the ER Forum 2017 and the ER 2017 Demo track*, 285-291.

Fernandez, E. B., Monge, R., & Hashizume, K. (2016). Building a security reference architecture for cloud systems. *Requirements Engineering*, *21*(2), 225–249.

Fernandez, E. B., Yoshioka, N., & Washizaki, H. (2008). Abstract security patterns. *Proceedings of the 2nd PLoP Workshop on Software Patterns and Quality (SPAQu'08).*

Fernandez, E. B., Yoshioka, N., & Washizaki, H. (2014). Patterns for cloud firewalls. *Proceedings of the 3rd Asian Conference on Pattern Language of Programs (AsianPLoP 2014).*

Fernandez, E. B., Yoshioka, N., & Washizaki, H. (2015). Cloud Access Security Broker (CASB): A pattern for accessing secure cloud services. *Proceedings of the 4th Asian Conference on Pattern Languages of Programs (AsianPLoP 2015).*

Fernandez, E. B., Yoshioka, N., & Washizaki, H. (2015). Patterns for Security and Privacy in Cloud Ecosystems. *Proceedings of the 2nd International Workshop on Evolving Security and Privacy Requirements Engineering (ESPRE 2015).*

Fernandez, E. B., Yoshioka, N., & Washizaki, H. (2016). Patterns for Secure Cloud IaaS. *Proceedings of the 5th Asian Conference on Pattern Languages of Programs (AsianPLoP 2016).*

Fernández, E. B., Yoshioka, N., & Washizaki, H. (2019). Using Security Patterns to Develop Secure Systems - Ten years later. *International Journal of Systems and Software Security and Protection*, *9*(4).

Fernandez, E. B., Yoshioka, N., Washizaki, H., Jurjens, J., VanHilst, M., & Pernul, G. (2010). Using security patterns to develop secure systems. In H. Mouratidis & I. G. I. Global (Eds.), Software Engineering for Secure Systems (pp. 16–31). Academic Press.

Fernández, E. B., Yoshioka, N., Washizaki, H., & Syed, M. H. (2016). Modeling and Security in Cloud Ecosystems. Future Internet, 8(2).

Fernandez, E. B., Yoshioka, N., Washizaki, H., & Yoder, J. (2014). Abstract security patterns for requirements and analysis of secure systems. *Proceedings of the 17th Workshop on Requirements Engineering (WER 2014).*

Fernández, E. B., Yoshioka, N., Washizaki, H., & Yoder, J. (2018). An abstract security pattern for Authentication and a derived concrete pattern, the Credential-based Authentication. *Proceedings of the 7th Asian Conference on Pattern Languages of Programs (AsianPLoP 2018).*

Hashizume, K., Yoshioka, N., & Fernandez, E. B. (2011). Misuse Patterns for Cloud Computing. *Proceedings of the 2nd Asian Conference on Pattern Languages of Programs (AsianPLoP'11).*

Hashizume, K., Yoshioka, N., & Fernandez, E. B. (2012). Three Misuse Patterns for Cloud Computing. In D. G. Rosado, E. Fernandez-Medina, & I. G. I. Global (Eds.), Security Engineering for Cloud Computing: Approaches and Tools. Academic Press.

Hazeyama, A. (2012). Survey on Body of Knowledge Regarding Software Security. *Proceedings of the 13th ACIS International Conference on Software Engineering, Artificial Intelligence*.

Islam, S., Ouedraogo, M., Kalloniatis, C., Mouratidis, H., & Gritzalis, S. (2018). Assurance of Security and Privacy Requirements for Cloud Deployment Models. *IEEE Transactions on Cloud Computing*, *6*(2), 387–400.

Ismail, U. M., & Islam, S. (2020). A unified framework for cloud security transparency and audit, Journal of Information Security and Applications. *Journal of Information Security and Applications*, *54*, 1–18.

Jelacic, B., Rosic, D., Lendak, I., Stanojevic, M., & Stoja, S. (2017). STRIDE to a Secure Smart Grid in a Hybrid Cloud. *Proceedings of the International Workshop on Security and Privacy Requirements Engineering (SECPRE)*.

Kalloniatis, C., Kavakli, E., & Gritzalis, S. (2008). Addressing privacy requirements in system design: The PriS method. *Requirements Engineering*, *13*.

Lenhard, J., Fritsch, L., & Herold, S. (2017). A literature study on privacy patterns research. In *Proceedings of the 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, (pp. 194-201). IEEE.

Martín, Y. S., & del Álamo, J. M. (2017). A metamodel for privacy engineering methods. *Proceedings of the CEUR Workshop, 3rd International Workshop on Privacy Engineering*.

Microsoft. (2002). *The STRIDE Threat Model*. https://msdn.microsoft.com/enus/library/ee823878(v=cs.20).aspx

MITRE. (1999). *Common Vulnerabilities and Exploits*. https://cve.mitre.org/

MITRE. (2012). *CVE-2012-4394*. https://cve.mitre.org/cgi-bin/cvename.cgi?name=2012-4394

Mouratidis, H., Shei, S., & Delaney, A. (2020). A security requirements modelling language for cloud computing environments. *Software & Systems Modeling*, *19*, 271–295.

Nhlabatsi, A., Bandara, A., Hayashi, S., Haley, C. B., Jurjens, J., Kaiya, H., Kubo, A., Laney, R., Mouratidis, H., Nuseibeh, B., Tahara, Y., Tun, T. T., Washizaki, H., Yoshioka, N., & Yu, Y. (2010). Security Patterns: Comparing Modeling Approaches. In H. Mouratidis & I. G. I. Global (Eds.), Software Engineering for Secure Systems (pp. 75–111). Academic Press.

Rath, A., Spasic, B., Boucart, N., & Thiran, P. (2018). Security Pattern for Cloud SaaS: From system and data security to privacy. *Computers*, *8*(2), 34–61.

Reimer, T., Abraham, P., & Tan, Q. (2013). Federated Identity Access Broker Pattern for Cloud Computing. *Proceedings of the 16th International Conference on Network-Based Information Systems (NBiS)*.

Riehle, D., & Zullighoven, H. (1996). Understanding and Using Patterns in Software Development. *Theory and Practice of Object Systems*, *2*(1), 3–13.

Schumacher, M., Fernandez, E. B., Hybertson, D., Buschmann, F., & Sommerlad, P. (2006). *Security Patterns: Integrating Security and Systems Engineering*. Wiley.

Subashini, S., & Kavitha, V. (2011). A survey on security issues in service delivery models of cloud computing. *Journal of Network and Computer Applications*, *34*(1), 1–11.

Tesoriero, R. (2011). Model-Driven Privacy and Security in Multimodal Social Media UIs. *Proceedings of the International Workshop on Modeling Social Media (MSM)*.

Washizaki, H. (2017). Security Patterns: Research Direction, Metamodel, Application and Verification. *Proceedings of the 2017 International Workshop on Big Data & Information Security (IWBIS).*

Washizaki, H., Fukumoto, S., Yamamoto, M., Yoshizawa, M., Fukazawa, Y., Ogata, S. B., Fernandez, E. B., Yoshioka, N., Kato, T., Kaiya, H., Kanuka, H., Kondo, Y., Okubo, T., & Hazeyama, A. (2016). A Metamodel for Security and Privacy Knowledge in Cloud Services. *Proceedings of the 12th IEEE World Congress on Services (SERVICES 2016)*.

Washizaki, H., Xia, T., Kamata, N., Fukazawa, Y., Kanuka, H., Yamamoto, D., Yoshino, M., Okubo, T., Ogata, S., Kaiya, H., Kato, T., Hazeyama, A., Tanaka, T., Yoshioka, N., & Priyalakshmi, G. (2018). Taxonomy and Literature Survey of Security Pattern Research. *Proceedings of the IEEE Conference on Applications, Information and Network Security (AINS).*

Xia, T., Washizaki, H., Kato, T., Kaiya, H., Ogata, S., Fernandez, E. B., Kanuka, H., Yoshino, M., Yamamoto, D., Okubo, T., Yoshioka, N., & Hazeyama, A. (2018). Cloud Security and Privacy Metamodel: Metamodel for Security and Privacy Knowledge in Cloud Services. *Proceedings of the 6th International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2018)*.

Yoshioka, N., Washizaki, H., & Maruyama, K. (2008). A survey on security patterns. *Progress in Informatics*, *5*, 35-47.

Tian Xia received M.S. degree in School of Computer Science and Engineering from Waseda University in 2017. He is working at DreamArts Corporation as an Engineer.

Hironori Washizaki received Doctor's degree in School of Computer Science and Engineering from Waseda University in 2003. He is now a Professor and the Associate Dean of the Research Promotion Division at Waseda University in Tokyo, and a Visiting Professor at the National Institute of Informatics. He also works in industry as Outside Directors of SYSTEM INFORMATION and eXmotion. Since 2017, he has been the lead on a large-scale grant at MEXT called enPiT-Pro SmartSE, which encompasses IoT, AI, software engineering and business.

Yoshiaki Fukazawa received Doctor's degree in School of Computer Science and Engineering from Waseda University in 1986. He is now a professor of Department of Information and Computer Science, Waseda University.

Haruhiko Kaiya is a professor at Kanagawa University, Hiratsuka, Japan.

Shinpei Ogata received his BE, ME, and PhD degrees from Shibaura Institute of Technology in 2007, 2009, and 2012, respectively. He is an associate professor at Shinshu University. His current research interests include model-driven engineering.

Eduardo B. Fernandez (Eduardo Fernandez Buglioni) is a professor in the Department of Computer Science and Engineering of Florida Atlantic University. He has published numerous papers as well as several books on computer security and software architecture. He holds a BS degree in Electrical Engineering from Universidad Tecnica Federico Santa Maria, Chile, an MS in EE from Purdue University, Lafayette, Indiana, and a Ph.D. in Computer Science from UCLA. He has published numerous papers on authorization models, object-oriented analysis and design, cloud computing, and security patterns. He has written four books on these subjects, the most recent being a book on security patterns. He is an active consultant for industry, including assignments with IBM, Allied Signal, Panasonic, Motorola, Lucent, Huawei, and others.

Takehisa Kato received Ph. D. in Informatics from Shizuoka University in 2013. He is currently an Engineer of Security Human Resource Management Dept. at Hitachi, Ltd.

Hideyuki Kanuka received the B.E. from Musashi Institute of Technology (currently Tokyo City University) in 2001 and the M.E. from Tokyo Institute of Technology in 2003. He is currently a Senior Researcher of Systems Innovation Center at Hitachi, Ltd.

Takao Okubo is a professor at Institute of Information Security. He received the MS degree in Engineering from Tokyo Institute of Technology in 1991. His current interests are secure development and threat analysis.

Nobukazu Yoshioka is an associate professor at National Institute of Informatics.

Atsuo Hazeyama received his Doctor's degree from Shinshu University in 1999. He is a professor at Tokyo Gakugei University. His current research interests include software security.