Cloud Security and Privacy Metamodel

Metamodel for Security and Privacy Knowledge in Cloud Services

Tian Xia[1], Hironori Washizaki[2], Takehisa Kato[3], Haruhiko Kaiya[4], Shinpei Ogata[5],
Eduardo B. Fernandez[6], Hideyuki Kanuka[7], Masayuki Yoshino[7], Dan Yamamoto[7], Takao Okubo[8],
Nobukazu Yoshioka[9] and Atsuo Hazeyama[10]

[1]Department of Computer Science and Communications Engineering, Waseda University, Japan
[2]Waseda University, Japan
[3]Toshiba Digital Solutions Corporation, Japan
[4]Kanagawa University, Japan
[5]Shinshu University, Japan
[6]Florida Atlantic University, U.S.A.
[7]Hitachi, Ltd., Japan
[8]Institute of Information Security, U.S.A.
[9]National Institute of Informatics, Japan
[10]Tokyo Gakugei University, Japan

Keywords: Cloud Computing, Metamodel Model, Security Patterns, Privacy Patterns, Software and System Architecture.

Abstract: Security and privacy are important in cloud services. Numerous security and privacy patterns as well as non-pattern-based knowledge such as practices and principles exist in cloud services. Selecting and combining the appropriate knowledge is difficult due to numerous options and the nature of the layered cloud stack. Herein we propose a metamodel called the Cloud Security and Privacy Metamodel (CSPM) to handle security and privacy in cloud service development and operations. CSPM can classify and support existing cloud security and privacy patterns and practices in a consistent and uniform manner. Moreover, we propose a security and privacy aware process to develop cloud system utilizing CSPM. Several case studies verify the effectiveness and usability of our approach. As a result, we confirmed effectiveness and usability of CSPM, as well as some possible future work.

1 INTRODUCTION

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 important. However, not all software engineers are experts on S&P, making it difficult to mitigate various S&P concerns throughout the software lifecycle.

A pattern is an abstraction from a concrete form that recurs in non-arbitrary contexts. Pattern catalogs (and pattern languages) should enable coherent integration and presentation of the relevant background, leitmotifs, and metaphors. Numerous S&P patterns have been published for cloud computing and services (D. Riehle, 1996) (K. Hashizume, 2011) (K. Hashizume, 2013). Moreover, non-pattern-based knowledge, which comes in the form of practices and principles, has been well documented to address S&P issues in cloud services.

The vast number of S&P patterns and documents describing related knowledge in cloud services makes selecting (and combining) the appropriate ones difficult. Although this is a common problem in security patterns in general, it is more severe in cloud services due to two reasons. Firstly, cloud services and their underlying mechanisms are related to various layers in the layered cloud stacks, and these services are often integrated over different layers. Secondly, a variety of devices are connected to cloud computing systems, which may require distinct

Xia, T., Washizaki, H., Kato, T., Kaiya, H., Ogata, S., Fernandez, E., Kanuka, H., Yoshino, M., Yamamoto, D., Okubo, T., Yoshioka, N. and Hazeyama, A.
Cloud Security and Privacy Metamodel - Metamodel for Security and Privacy Knowledge in Cloud Services.
DOI: 10.5220/0006600403790386
In *Proceedings of the 6th International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2018)*, pages 379-386
ISBN: 978-989-758-283-7

deployment models and diverse services, resulting in a highly complex system. This intertwined system leads to many concerns, including S&P.

Metamodels or reference architectures that capture the essential concepts related to S&P in layered cloud stacks should address the aforementioned problem since engineers can describe S&P-related knowledge as well as design systems and services consistently over many layers. Although several metamodels and abstract reference architectures address cloud security (Kleopatra Chatziprimou, 2013)(E. B. Fernandez, et al, 2015) (A. Hazeyama, 2012), none addresses privacy in cloud services. Since the relation between S&P is complex, it is preferable to deal with S&P simultaneously. On the other hand, several metamodels and conceptual models address both S&P (C. Kalloniatis, E. Kavakli, and S. Gritzalis,2008), but they are generally defined in such a way that makes applying them directly to cloud services challenging.

We previously presented the background and former version of the metamodel (H. Washizaki, et al., 2016). In this research, we propose an extended metamodel called the “Cloud Security and Privacy Metamodel (CSPM)” to address S&P in cloud services by integrating and extending existing cloud security metamodels together with newly added concepts.

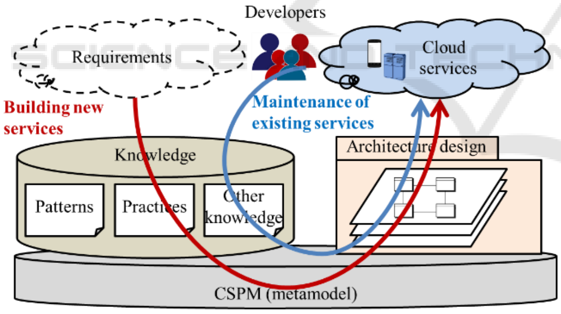


Figure 1: Overview of cloud services and our metamodel.

Figure 1 shows how to use CSPM in cloud service development and maintenance. CSPM provides the basis to describe and accumulate S&P-related knowledge over multiple layers, making it easier to select and combine the appropriate patterns and related knowledge to address S&P issues in cloud services. Moreover, engineers and developers can refer to CSPM to design high-level architectures of cloud service systems efficiently and effectively.

As an extension to our previous research, we conducted several case studies to address some following questions:

- RQ1. Does CSPM help developers address S&P problems and the corresponding patterns?
- RQ2. Do the S&P solutions by CSPM effectively improve the system?
- RQ3. Are CSPM and the process using CSPM practical in real cloud system development and maintenance?

RQ1 and 2 focuses on the effectiveness of CSPM from different view point. RQ3 discuss the usability of our approach, for both metamodel itself and the process we proposed.

The remainder of this paper is organized as follows. First, we propose our metamodel in Section 2. Section 3 describes our case studies and discusses the results with respect to the RQs. Finally, we conclude our work and discuss the future direction in Section 4.

2 CSPM

2.1 Problems to Be Addressed

Consider the scenario where a developer who is new to cloud development is tasked to build a cloud application. As he is not an expert, he prefers to have some documents about security and privacy. However, searches for such documents lead to several problems:

- **Vast number of S&P patterns and documents:** A pattern is a general reusable solution to a reoccurring problem. Numerous S&P patterns as well as existing documents applicable to cloud development exist. Hence, selecting the appropriate patterns from a huge knowledge base is extremely challenging. Additionally, many S&P patterns are not specific to cloud services (E.B. Fernandez, et al., 2010). Utilizing non-S&P specific patterns may be complicated and burdensome.
- **Complex relationships between a cloud service and its mechanism:** A cloud can typically be divided into three layers: infrastructure, platform, and software. From the user’s viewpoint, each service is provided at a certain layer. However, the data controlled by the service may be related to any layer (S. Subashini and V. Kavitha, 2011). For example, a cloud storage service containing hardware and cluster controller belongs to infrastructure for storing data. At the same time, the interface in the software layer for a user is required. Just this

situation makes selecting and utilizing patterns and documents hard.

- **No practical metamodel is applicable:** Existing metamodels (C. Kalloniatis, E. Kavakli, and S. Gritzalis,2008) could be good references because they contain essential concepts when dealing with S&P issues. However, there is not a metamodel can practically deal with the S&P problem in real cloud development.

2.2 Requirements and High-Level Architecture of the Metamodel

According to problems described in Section 1 and 2.1, we identified three requirements for designing a metamodel:

- R1. The metamodel must consistently deal with S&P-related knowledge over multiple layers, including the software application layer, the platform layer, and the infrastructure layer. Services corresponding to these layers are SaaS (Software as a Service), PaaS (Platform as a Service), and IaaS (Infrastructure as a Service). Cloud services are often integrated over different layers. Thus, security of multiple layers must be carefully considered (A.A. Almutairi, et al., 2012). This is also important for privacy.
- R2. The metamodel must be mostly consistent with existing cloud security metamodels and reference architectures. Engineers and developers must be able to utilize assets based on our metamodel and those based on existing metamodels (and reference architectures).
- R3. The metamodel allows engineers and developers convenient access to a knowledge base containing cloud-specific and cloud-independent knowledge.

Based on these requirements, we designed CSPM to consist of seven packages.

2.3 Design of the Metamodel

Figure 2 shows the details of CSPM as a UML class diagram. Table 1 describes the outline and major concepts of these packages. The metamodel satisfies the above requirements as follows:

- The problem, bridge, and solution packages capture concepts common to all layers and organize their relationships. These packages are used as foundation for all layers, and provide consistent handling of S&P-related knowledge over different layers. This satisfies R1.
- The aforementioned common packages incorporate most of the concepts using the relationships defined in existing metamodels (E. B. Fernandez, et al., 2015) (A. Hazeyama, 2012). Hence, the entire metamodel is mostly consistent with existing metamodels. This satisfies R2.
- By separating general concepts in the problem, bridge, and solution packages from those specific to a certain layer, cloud-specific knowledge and cloud independent knowledge are easier to access. This satisfies R3.

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	Application, coding rule	
Platform	Concepts specific to the platform layer	Virtual environment, virtual storage	
Infrastructure	Concepts specific to the infrastructure layer	Virtual machine, hardware	
Target	Concepts specific to the target application	Goal, policy, asset, cloud service	

2.4 Usage of CSPM

CSPM declares almost everything for a cloud system. It has a large scale, which may be difficult to implement in real developments. The lack of descriptions for detailed usage is one reason metamodels seem impractical. Herein we propose a process for S&P development with CSPM and show its usage from various viewpoints with an emphasis on components in the metamodel.

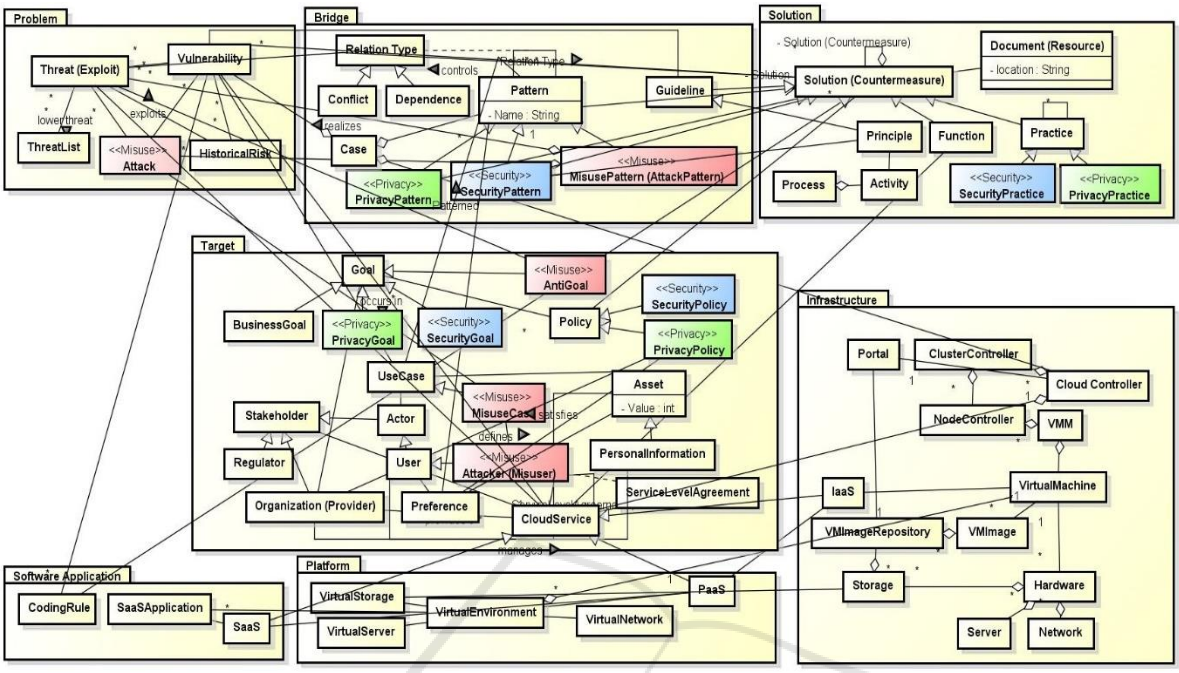


Figure 2: Overview of CSPM.

2.4.1 S&P Development Process

Figure 3 shows the suggested process for S&P development:

①. S&P Requirement Analysis: The system requirements should be analyzed first. Based on these requirements, developers must determine threats in the current system model using a threat model such as STRIDE (Microsoft, 2002). As described later, CSPM in Vulnerability View can also support this step.

②. S&P Design: After analyzing S&P problems in step 1, developers must identify solutions. As mentioned at beginning in this paper, S&P patterns as well as some documents contribute to this step because they suggest a solution to the system. Pattern View, which is discussed below, can assist developers in selecting and using patterns from the knowledge base.

③. S&P Implementation & Test: After revising the system, developers can implement their system and test. If new problems arise, developers can return to step 1 and repeat this process. In some cases, which combine several patterns to derive an original solution, the process itself could be added to the knowledge base.

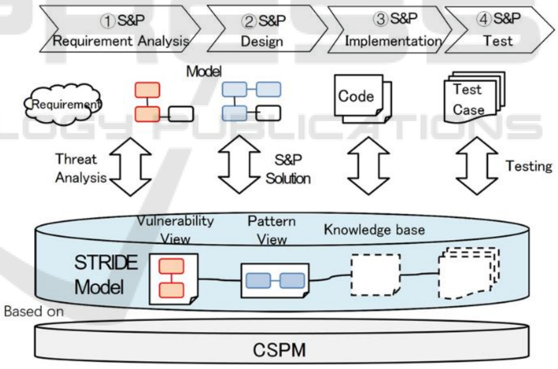


Figure 3: Overview of S&P Development Process.

2.4.2 Vulnerability View

Figure 4 shows an example of the model in Vulnerability View. This model is simplified from the metamodel. That is, it is an instance of the metamodel. This view can be used to model vulnerabilities from databases such as the Common Vulnerabilities and Exposures (CVE), which is a well-known database that provides a dictionary for publicly known information-security vulnerabilities and exposures.

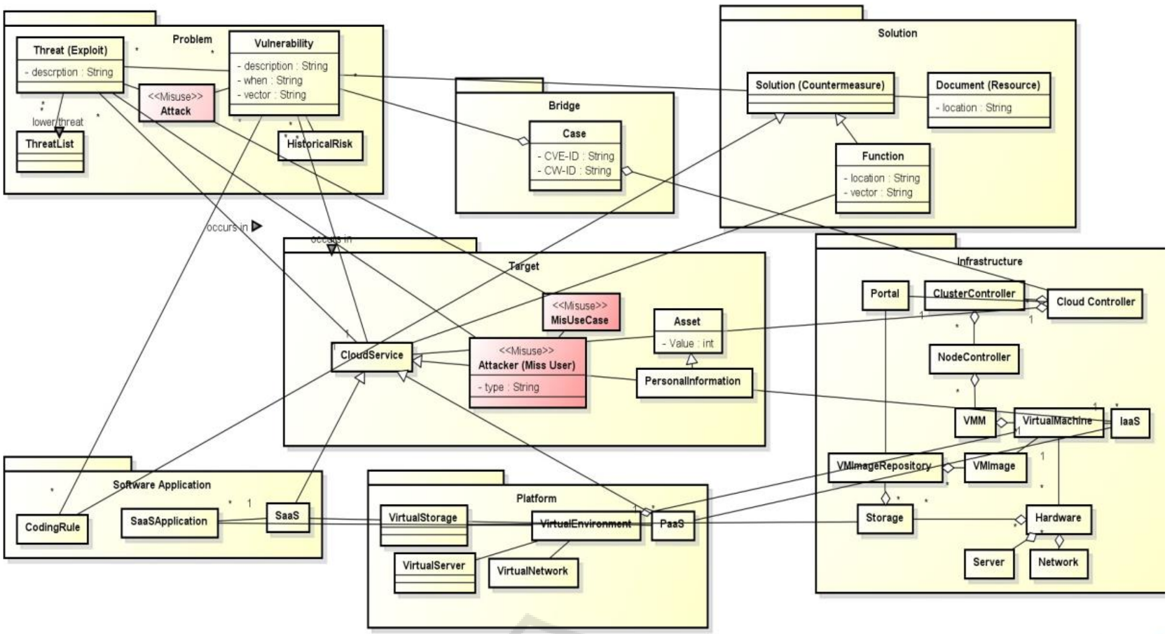


Figure 4: Overview of the Vulnerability View model.

2.4.3 Pattern View

Figure 5 shows an example of a simplified metamodel in Pattern View. This model mainly focuses on components related to S&P patterns such as goals, threats, and solutions. This model makes it easier to analyze the requirements and threats to the system. Thus, S&P patterns to improve the system can be identified. An example is presented as a case study in Section 3.

We also suggest representing the pattern problem and solution text using this model. Figure 6 shows an example where the pattern text explains the problem and solution well. Modeling of a pattern may also be meaningful for development, especially for developers unfamiliar with S&P patterns.

3 EXPERIMENT AND CASE STUDY

3.1 Contrast Experiment

To confirm the contribution of CSPM and answer to the RQs, we conducted a contrast experiment.

3.1.1 Experiment Preparation

The experiment had ten participants. All were college students ranging from fourth year undergraduate to second year master students. Participants were divided equally into two groups: experiment group (EG) and control group (CG).

For both groups, participants were assigned a system model simplified from a student work, which contained several security threats. They were initially asked to read the class diagram as well as use case explanation to identify the S&P problems existing in the system. Participants were also expected to solve the identified problems on the model level. We prepared some S&P patterns as a reference, but not all patterns were useful in this system. After they finished, they completed a questionnaire.

3.1.2 Experiment Results

Figure 7 is a box plot represents the experiment data to show the distribution. We measured three variables: number of problems found in the system, number of problems solved by revising the model, and number of patterns used to solve problem.

According to figure 7, although we failed to confirm difference in identifying problems, EG was more proficient at revising. All participants in EG revised at least three main S&P problems, whereas the number of revision varied widely in CG. We attribute this to the usage of S&P patterns in EG.

We hypothesized that EG would complete the tasks faster than CG. However, the result could not show the difference between the two groups. One reason is that EG spent more time reading the metamodel and guideline.

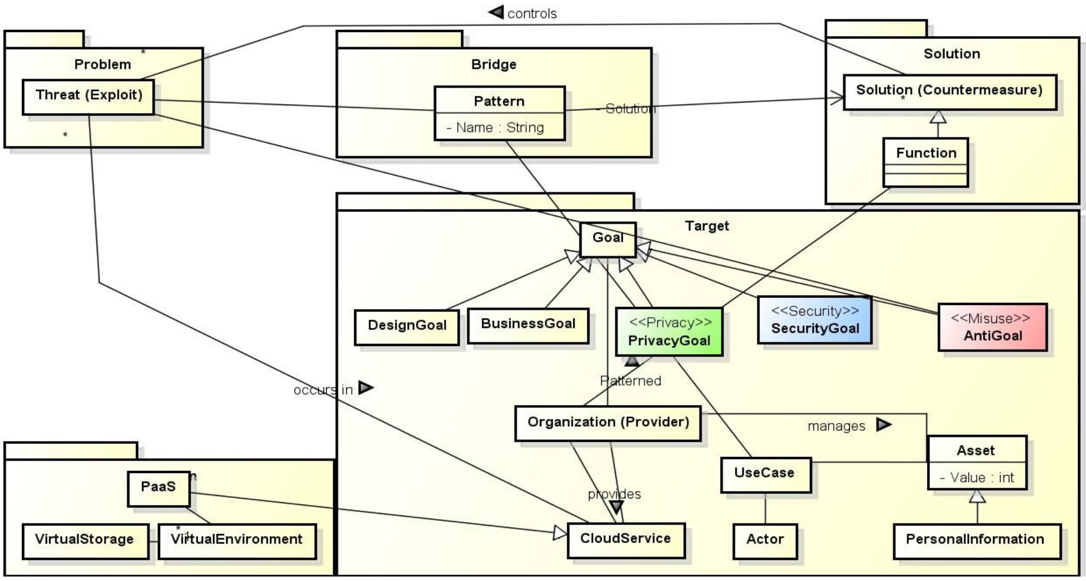


Figure 5: Overview of the Pattern View model.

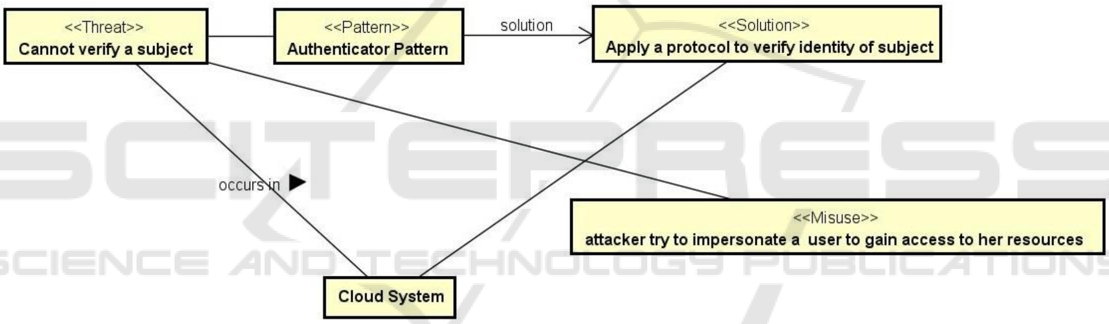


Figure 6: Example of the Pattern View structure using the Authenticator Pattern.

The questionnaire asked EG participants about their opinion of CSPM. Interestingly, all participants gave similar answers for each question. They declared that the Pattern View of the metamodel itself (Figure 5) is easily understood, but not very useful. However, the explanation and example in the guideline are very helpful, especially for applying patterns. Participants also said the Pattern View structure of the S&P pattern is very helpful, but that they prefer this diagram along with a detailed description of the patterns.

3.2 Case Study: “Treasure-Hunting Game”

We conducted a case study to confirm the effectiveness of the solutions analyzed by CSPM. This case study is based on an Android game application, which uses cloud for data storage. We expect to confirm the contribution by comparing the original version of the system with the one improved by our metamodel.

3.2.1 Preparation

In a normal development, S&P analysis is conducted prior to designing the model. But in our study, to confirm the contribution of CSPM, a student work designed without S&P analysis was used. Amazon Web Service (AWS), was used in the case study for implementing cloud function.

3.2.2 Results

S&P requirement is analyzed based on the STRIDE model. Some threats such as tampering with local data or listening to transmissions are addressed by Android API and AWS API. Thus, we mainly focused on the following two topics:

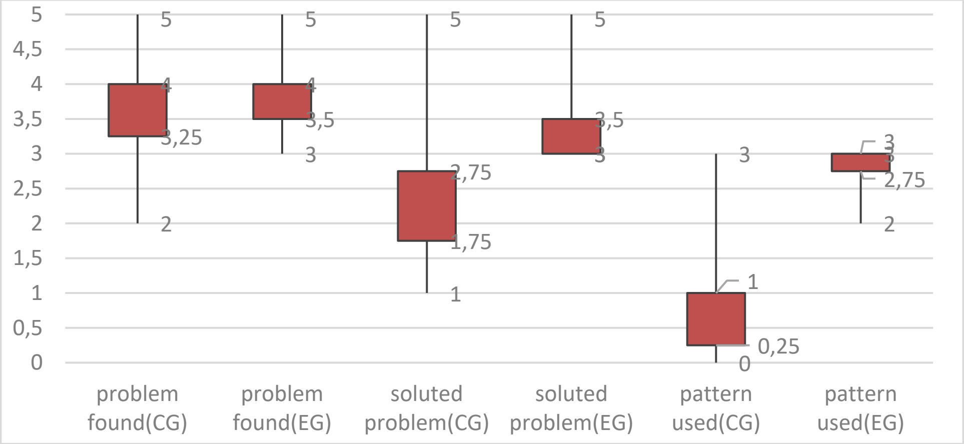


Figure 7: Box plot of data in the case study.

- **Authentication Problem**: This system has a high risk of identity spoofing due to the lack of authentication.
  Pattern and Solution: Based on the *Authenticator Pattern*, an authenticator is added. The *Authenticator Pattern* requires a user to sign up and sign in before accessing to the system.
- **Access Right Problem**: The function of checking user data requires for a user's name. If the user's data exist, it appears on the screen. This function might be designed to allow a user check a friend's data. However, everyone can check each other's information.

Pattern and Solution: Before accessing cloud storage data, we added an access right controller according to the *Role-based Access Control Pattern*.

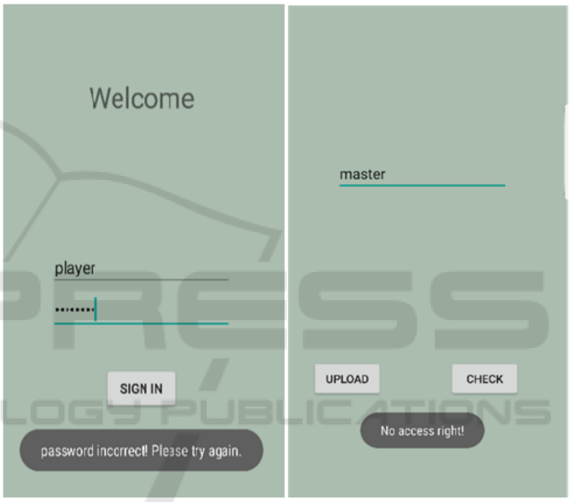
Figure 8 shows the S&P problems are revised by implementing patterns.

3.3 Discussion

- **RQ1. Does CSPM help developers address S&P problems and the corresponding patterns?**

The contrast experiment confirmed that the experiment group solved more problems in the same or even less time. Due to the support of our approach, all the participants in EG were able to select the appropriate pattern from the knowledge base and use it to revise the model. We expect the same or even better behaviors can be expected when dealing with larger scale of system which has more S&P issues.

The benefits of the Pattern View structure of the S&P pattern were also noted. It can help developers not only identify the necessary pattern, but also improve understanding of the patterns.



password incorrect! Please try again.

No access right!

Figure 8: Result after implementing patterns.

In addition, as we mentioned at beginning of this paper, existing research on metamodel was focused on security issue only. In our study, case study shows that both security (i.e. authentication) and privacy (i.e. access right control) issues can be solved by CSPM.

With regard to the combination of S&P patterns, we only confirmed some simple combinations due to the small scale of the target system. Further experiments on a complex system are necessary.

The contribution of determining problems from systems has yet to be confirmed because the Pattern View mainly focuses on dealing with the S&P pattern, and patterns are typically utilized after identifying problems. Further experiments such as a case study should be conducted.

The results confirm that CSPM can support developers in selecting and using S&P patterns to solve S&P problems by the Pattern View. A future study should verify the contribution on finding problems in a more complex system and combining several S&P patterns.

- **RQ2. Do the S&P solutions by CSPM effectively improve the system?**

The case study examined the results of Pattern View of CSPM to revise the model. We implemented an application based on this model as well as the original model. We confirmed that the problems found in the target system exist are resolved in the revised version. Because this case study focused on a simple system, some components of the cloud system in CSPM were not considered in this study. It is possible that additional issues arise in a more complex system.

The effectiveness of the solution of CSPM for a simple system is confirmed. Additional case studies are necessary to evaluate the whole metamodel.

- **RQ3. Are CSPM and the process using CSPM practical in real cloud system development and maintenance?**

Both the contrast experiment and the case study followed the proposed process. The results indicate that CSPM is practical. The contrast experiment confirmed that some detail usages of CSPM proposed in this paper are applicable to S&P analysis, while the case study showed an example of cloud system development with CSPM.

The participants in EG had some negative comments. They expressed that the metamodel itself is not useful. They declared that the guideline is much more helpful than the model itself. A revised guideline that includes more usages of CSPM should make our approach more practical.

Thus, CSPM is practical in some usages. In the future, creating a guideline to describe its usages should increase the applicability of CSPM.

4 CONCLUSION AND FUTURE WORK

We proposed a metamodel, CSPM, to address security and privacy in cloud services as well as a process to use CSPM in development. Two case studies were conducted to verify the effectiveness and usability of CSPM.

In the future, we plan to conduct more complex case studies such as designing a cloud system containing several layers based on CSPM and implementing them to evaluate our approach. We also plan to utilize the Vulnerability View and the Pattern View via a semi-automatically applied pattern if a specific threat is detected. Another future work will involve developing detailed guidelines for CSPM to expand its usage, allowing more developers, especially newcomers, to apply our approach.

ACKNOWLEDGEMENTS

We are grateful to the participants in the contrast experiment.

REFERENCES

D. Riehle and H. Zullighoven, 1996, “Understanding and Using Patterns in Software Development,” Theory and Practice of Object Systems, Vol.2, No.1, pp.3-13.

K. Hashizume, N. Yoshioka and E.B. Fernandez, 2011, “Misuse Patterns for Cloud Computing,” 2nd Asian Conference on Pattern Languages of Programs (AsianPLoP’11).

K. Hashizume, N. Yoshioka and E.B. Fernandez,2013, “Three Misuse Patterns for Cloud Computing,” in “Security Engineering for Cloud Computing: Approaches and Tools,” IGI Global.

E. B. Fernandez, et al, 2015, “Building a security reference architecture for cloud systems,” Requirements Engineering Journal

A. Hazeyama, 2012, “Survey on Body of Knowledge Regarding Software Security”, 13th ACIS International Conference on Software Engineering, Artificial Intelligence

Kleopatra Chatziprimou, Kevin Lano, and Steffen Zschaler, 2013, "Towards a Meta-model of the Cloud Computing Resource Landscape." MODELSWARD.

C. Kalloniatis, E. Kavakli and S. Gritzalis,2008, “Addressing privacy requirements in system design: the PriS method,” Requirements Engineering, Vol.13

H. Washizaki, et al., 2016, “A Metamodel for Security and Privacy Knowledge in Cloud Services,” Proc. 12th IEEE World Congress on Services (SERVICES 2016)

S. Subashini and V. Kavitha, 2011, “A survey on security issues in service delivery models of cloud computing,” Journal of Network and Computer Applications, Vol.34, No.1, pp.1–11, 2011.

A.A. Almutairi, et al., 2012, “A Distributed Access Control Architecture for Cloud Computing,” IEEE Software, Vol. 29, No. 2, pp.36-44

E.B. Fernandez, N. Yoshioka, H. Washizaki, et al., 2010, “Using security patterns to develop secure systems”, in “Software Engineering for Secure Systems”, IGI Global, pp16-31

L.L. Lobato, E.B. Fernandez and S.D. Zorzo, 2009, “Patterns to support the development of privacy policies”, International Conference on Availability, Reliability and Security (ARES’09).