6-4 Description and Negotiation Techniques to Establish Security SLA

Takeshi TAKAHASHI

The security features of current online services are mostly defined by the service provider. Users may feel unhappy if the terms do not meet the expected criteria, since they can only agree to user the service with the terms or decline using the service. Even a simple negotiation might result in a more satisfying outcome in many cases. This article proposes a mechanism to build non-repudiable security service level agreements between a user and a service provider.

1 Introduction

In recent years, online services have greatly increased and developed, but at the same time cyber society has been facing an increase in the number of incidents of security and privacy threats. There is a need for mechanisms that protect users from such incidents, and various technical countermeasures already exist, but implementing all of them is not desirable from the viewpoints of cost and convenience of services. Therefore, the balance between security and convenience needs to be considered, but the environment and security requirements differ for each user, so determining a uniform balance would be unrealistic. The balance needs to be determined separately for each user or each time a service is used. There are prominent technical issues involved for determining this balance:

a) The user's security requirements must be described in a machine readable and structured format.
b) Regular users with limited technical knowledge find it very difficult to determine what are the mandatory security technologies, so technology by which such users can determine the security requirements is required.
c) An auto-negotiation technique is required for constructing a security policy which the service should satisfy. Currently, users have only two options of whether to agree or disagree with the security policy displayed by the service provider, and have no means for negotiating with the provider for necessary security levels or techniques. Then, even if the provider does consider negotiating with the users, manpower costs make it unrealistic from a cost viewpoint.
d) The terms of agreement generated as the result of the negotiations must assure non-repudiability. Even if the users and provider agree to a security policy that should be fulfilled, it does not mean that security related incidents will not occur. Therefore, if such an incident does occur and the cause of the incident is in breach of the terms of agreement, then the terms of that agreement can be used as grounds for the affected party to complain against the offender.

In order to deal with the above mentioned problems, and achieve a balance between security and convenience, a method for constructing a security SLA (SSLA) that will assure non-repudability is proposed. SSLA refers to an agreement between the service provider and its users that documents what should be the security level provided by the service provider. In this proposed method, as elemental technology, a security expression technique and ID conversion technique are provided.

With the security expression technique, the security requirements and capabilities can be described in a machine readable format. These security requirements and capabilities can be described from multiple viewpoints, and each viewpoint is known as a dimension. With the ID conversion technique, the information described from the point of different dimensions can be converted into the information based on any one dimension. With this method, users with limited technical knowledge will be able to describe the security requirements without using technical terminology, and automatic conversion to technical terminology is possible. Based on this, a user and the service provider will be able to negotiate for formulating the SSLA. The proposed method constructs an SSLA that assures that the result of the negotiations will be non-repudiable. As a result, users who till now could respond only

either yes or no to the security policy displayed by the provider, can now draw up a mutually agreeable security policy.

For detailed information, refer to reference documents [1][2], as this paper is a summary of those documents.

2 Summary of architecture

The proposed method defines three roles: User, Service Provider (SP) and Knowledge Base (KB). The user uses the online services, and the SP provides those services to the user. KB holds various types of information related to security, and also a dictionary which is needed for translating.

Figure 1 is a summary of the process of the proposed method. The user describes multiple security requirements in any number of dimensions. Using the ID conversion technique, the user converts and summarizes these requirements into a single dimension. By matching the User's security requirements with the capabilities of the SP, and negotiating, this constructs a security level that the services should satisfy, i.e. an SSLA.

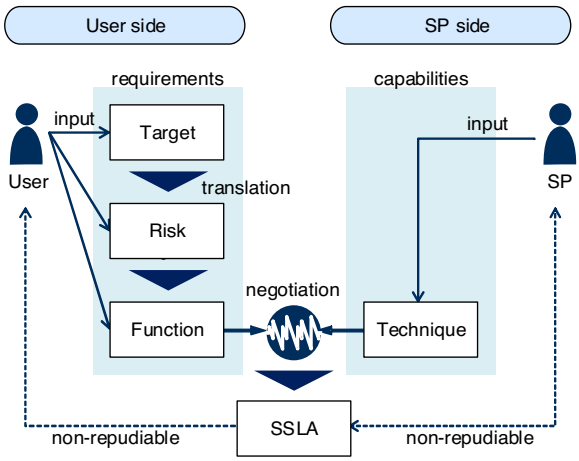

Fig. 1 Summary of process

3 Security expression technique

SSLA is the information on security level agreed between the User and SP. To construct an SSLA, the security requirements and capabilities must be clearly stated. "Security requirements" is information that writes what kind of security or security technology is required, and "capabilities" is information that writes what kind of technologies the SP has or what can be achieved. These two types of information are written using the vocabulary of the dictionary in the KB. In order to achieve machine processing, the proposed method aims at minimizing unstructured writing, so a unique identifier is assigned to each word. This identifier is expressed in the format of an Object Identifier (OID)[3]. Then, the SSLA is constructed by matching this security requirement and capabilities between the User and SP.

The proposed method has provided four dimensions: Target, Risk, Function, and Technique, so various users can freely describe the security requirements and capabilities. Then based on each dimension, a dictionary is provided that contains terms and identifier corresponding to each term. The Target dimension specifies the targets to be protected. This is described by selecting from the terms written in the Target dictionary, for example "Personal

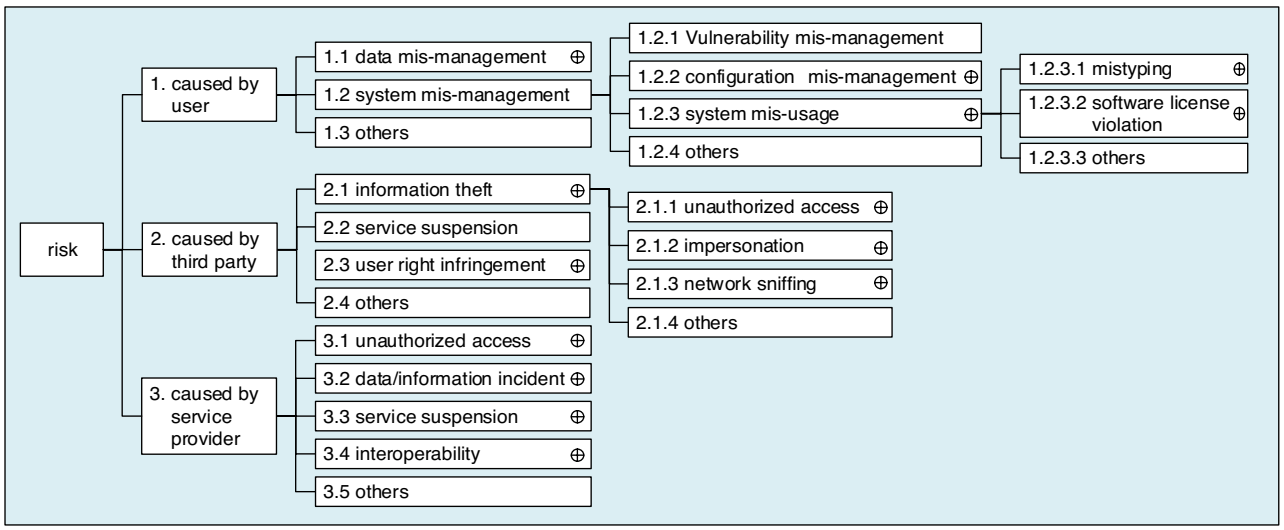

Fig. 2 Risk dictionary example (extract)

information" of the user. The Risk dimension specifies the types of risk that should be avoided. This is described by selecting from the terms written in the Risk dictionary, for example the risk of "Communication interception". The Function dimension specifies the functions that should be implemented. This is described by selecting from the terms written in the Function dictionary, for example "Encrypt user data" or "User authentication". The Technique dimension specifies the security technology/tools that should be implemented. This is described by selecting from the terms written in the Technique dictionary, for example "AES" or "SHA".

The identifier of each term begins with the OID arc of either TARGET, RISK, FUNCTION, or TECHNIQUE, for each respective dimension. Figure 2 shows a sample extract from the Risk dictionary. In each dictionary, the identifier of each item is described in the sequence after the OID arc of TARGET, RISK, FUNCTION, or TECHNIQUE. Each KB can also have its own dictionary. The User and SP usually have multiple security requirements and capabilities, so the security requirements and capabilities are actually expressed as the list of these OIDs.

Depending on the situation or depending on the User, there are cases where security is expressed using the terms of multiple dimensions. In such cases, the security can be described by combining the terms of multiple dimensions using a colon. For example, when Function corresponding to a particular Risk is specified, the terms of Risk and Function can be connected using a colon. For example, it can be written as "Risk.1.1.2:Function.19.12.2".

4 ID conversion technique

With multiple dimensions being provided for describing security requirements, the User can specify the security requirements from various viewpoints. As a result, the risk of not being able to specify important security requirements can be reduced. However, in order for the computer to automatically process different dimensional information, a technique for translating that information into any dimension is required.

The proposed method provides a translation mapping, which shows which OIDs of a particular dimension correspond to which OIDs of another dimension. The translation is done by referring to this translation mapping. This mapping is also saved in the KB. There are three types of mapping: [target, risk], [risk, function], and [function, technique]. These mappings are comprised of two columns. The OID of one column corresponds to multiple OIDs of another column. For example, the [risk, function] mapping is comprised of the OID that represents Risk and one or more Functions that correspond to it. One or more Functions are linked to a single Risk because there are actually cases where multiple functions are required for dealing with a particular risk.

5 Negotiation protocol

The proposed method carries out two types of communication: KB reference, and SSLA negotiation. KB reference is the procedure for translating security requirements and capabilities that are expressed in various dimensions, and SSLA negotiation is the procedure for constructing an SSLA that is agreeable between both parties.

KB reference starts when the query sender sends the information on security requirements and capabilities to the KB. The KB that received the information converts the dimension for the security requirements and capabilities, and returns that result to the query sender.

SSLA negotiation constructs SSLAs using SSLA-proposal and SSLA-confirmation messages. SSLA-proposal has the security requirements and capabilities, and if the receiver of the SSLA-proposal message agreed to the proposed security requirements, then the SSLA-confirmation is sent. If the receiver does not agree to the contents, then instead of sending the SSLA-confirmation, a new SSLA-proposal with different security requirements and capabilities is sent back. This procedure will continue until either the SSLA-confirmation is sent or the negotiations are discontinued.

When the SSLA-confirmation message arrives, the negotiations end, and the list of security requirements at that time will be the SSLA.

As mentioned above, the proposed method allows conversions of messages multiple times. To simplify the discussion, Fig. 3 shows an example where the negotiation procedure ends in one round. Here, KB_U is the KB that User trusts, and KB_{SP} is the KB that SP trusts. Before starting the negotiations, User contacts the KB_U and converts the security requirements that are described in various dimensions into security requirements in the Function dimension. User sends to SP the SSLA-proposal message containing the conversion result and the URI of KB_U. When SP receives the message, SP itself will check whether the proposed security requirements are satisfactory or not, and does not want to agree to the Function dimension with still

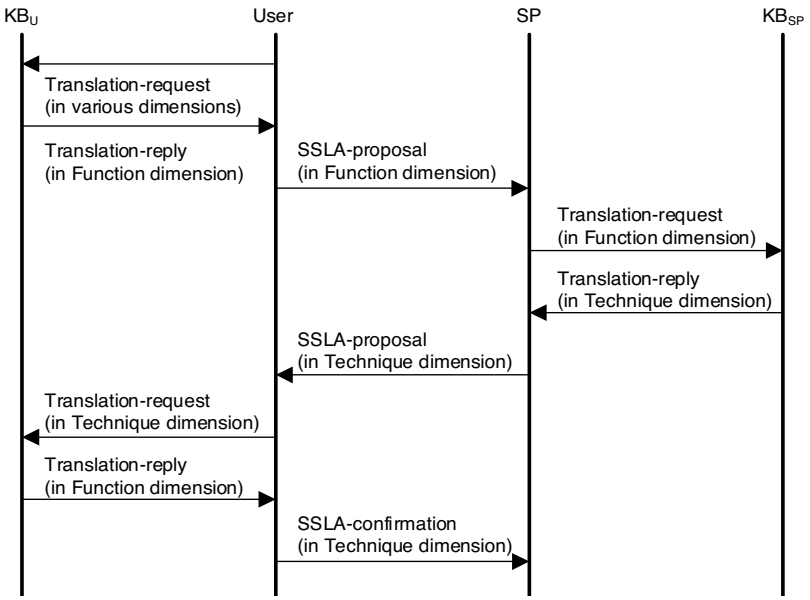

Fig. 3 Negotiation procedure example

vague security requirements; SP only wants to agree to the more specific Technique dimension, limiting SP's scope of responsibility. Therefore, SP will communicate with the KB_{SP}, and inquire about the list of security requirements in the Technique dimension, that meet the security requirements received from User. Then, a new SSLA-proposal message containing that inquiry result and the URI of KB_{SP} is constructed and sent to User. User received this proposal message, and once again searches the KB_U in order to confirm from the list of proposed Techniques, whether or not the security requirements in the Function dimension that he/she could not originally implement are satisfied. User sends to the KB_U the list of security requirements of the Technique dimension that were received from SP. User receives the list of security requirements that were converted to be based on Function dimension, so based on those security requirements, Use can confirm whether or not his/her own original security requirements are satisfied. After it is confirmed that there are no problems, User sends an SSLA-confirmation message to SP, and the SSLA is finally agreed to. Keep in mind that, at the beginning of this procedure, User sends the security requirements in the Function dimension, but the SSLA agreed to at the end is in the Technique dimension.

In the proposed method, to assure non-repudiability of the SSLA generated as a result of the negotiation, an encrypted identifier and digital signature are used in the message of the negotiation protocol. For details, refer to document [2].

6 Conclusion

The proposed method is able to construct an SSLA that assures non-repudiability by using a security expression technique, ID conversion technique, negotiation protocol, and SSLA determining algorithm. The effectiveness of the proposed method is shown from the viewpoints of feasibility, non-repudiability and DoS resistance (refer to document [1][2]), but the various techniques mentioned in this paper need to be further developed in the future through research. For example, regarding the way in which the security requirements and capabilities are to be described, it will be time consuming and troublesome for the user him/herself to specify, even if several dimensions are provided and he/she has knowledge. Accordingly, a method is needed that takes into account the situation and user, and automatically describes the security requirements and capabilities. This is especially important for items with small screens such as mobile phones, or where convenience is limited. We hope that by advancing this research, we can arrive at a stage in the future where the balance between security and convenience can be optimized for each user.

Acknowledgments

We wish to extend our heartfelt gratitude to Dr. Koji Nakao, Senior Researcher and Dr. Kazumasa Taira, Research Center Director, for their support in conducting this research.

References

1 T. Takahashi, J. Harju, J. Kannisto, B. Silverajan, J. Harju , S. Matsuo, "Tailored security: building nonrepudiable security service level agreements," IEEE Vehicular Technology Magazine, 2013.

2 J. Kannisto, T. Takahashi, J. Harju, S. Heikkinen, M. Helenius, S. Matsuo, B. Silverajan, "A Non-repudiable Negotiation Protocol for Security Service Level Agreements," International Journal of Communication Systems, 2015.

3 International Telecommunications Union, "Information technology - Open Systems Interconnection - Procedures for the operation of Object Identifier Registration Authorities: General procedures and top arcs of the International Object Identifier tree," ITU-T Recommendation X.660 , 2011.

Takeshi TAKAHASHI, Ph.D.

Senior Researcher, Cybersecurity Laboratory,
Cybersecurity Research Institute
Cybersecurity, Network Security