International Journal of Security and Its Applications
Vol.7, No.5 (2013), pp.305-314
http://dx.doi.org/10.14257/ijsia.2013.7.5.28

Access-control-based Efficient Privacy Protection Method for Social Networking Services

Yu-Jong Jang[1] and Jin Kwak[2]

[1]*ISAA Lab, Department of Information Security Engineering, Soonchunhyang University, Korea*
[2]*Department of Information Security Engineering, Soonchunhyang University, Korea*
yjjang@sch.ac.kr, jkwak@sch.ac.kr

Abstract

There has been a recent surge in the popularity of social networking services (SNSs) and SNSs have grown rapidly, as has the variety of information shared through SNSs. However, SNSs raise concerns about the security and privacy of users because the information written by the user might be exposed in the SNS. Many studies have addressed this issue, but previous research has lacked methods that can be applied efficiently in the SNS environment. In this paper, we propose a method for the efficient protection of privacy in SNS based on access control

Keywords: *social network service, hash chain, ticket, access control*

1. Introduction

Social networking services (SNSs) allow users to connect via a network. They make it easy to share a variety of information and SNSs have developed rapidly through information sharing. However, SNS have security problems related to privacy where information might be shared with non-friends. Many studies have investigated SNS security based on data encryption and user authentication. However, these studies have not solved the security problems of information leakage and unwanted information sharing. Previous studies have lacked methods that can be applied efficiently in the SNS environment[1, 8].

In this paper, we propose a hash-chain-based access control information leakage prevention scheme, which can avoid the unwanted sharing of information with other users.

2. Related Work

2.1. SNS Architecture

The SNS architecture comprises a large network of convergent individuals in a network. This architecture is depicted as a line graph among users in Figure 1.

Figure 1 shows the relationships among users (A, B, C, D, E, F, G, and H) in an SNS architecture. The friend relationships are direct connections among users in the graph, *e.g.*, (A, B), (B, C), and (G, D). Non-friend relationships are shown as indirect connections to a user in the graph. Thus, non-friend relationships are connected through a friend, *e.g.*, (B, E : B–A–E) and (D, C : D–B–C)[9].

ISSN: 1738-9976 IJSIA

It is possible to express the overall SNS architecture via users as a single unit. It is also possible to analyze the relationships between users based on the distance of separation in the object graph (Table 1).

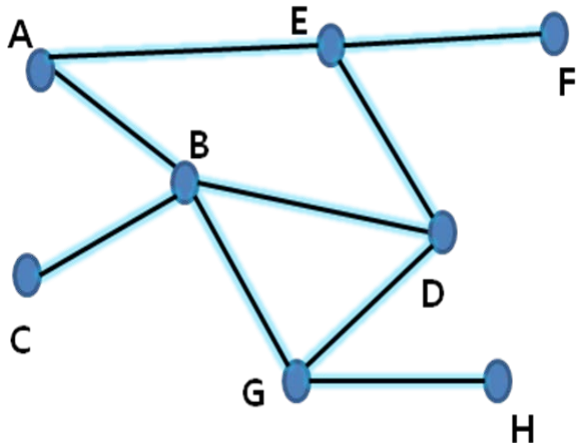

Figure 1. SNS Architecture Graph

Table 1. SNS Friend Relationships

User	Distance from D
B, G, E	1
H, F, A, C	2

Table 1 shows an analysis of the relationships with user D according to the graph shown in Figure 1. Friend relationships have a distance of 1 in the graph of user D because the user is connected directly to user D. Non-friend relationships have a distance of 2 in the graph of user D because the user is connected indirectly to user D via a friend relationship[2, 10].

2.2. Threat of SNS Privacy Leakage

SNS was created for information sharing. SNS users share information with groups of friends. Thus, SNS users access the information in their group of friends but their information may be exposed to non-friends. Thus, publicly available information may be exploited for malicious purposes. This is known as privacy exposure.

Figure 2 shows the information leakage exposure process. Information leakage occurs through a friend [3, 4]. Figure 2 shows an example of information leakage over Facebook.

The contents of the conversation between User A and User B are leaked to User E. User E does not have a friend relationship with User B; however, User A has a friend relationship with User E. Hence, User E can see User B's information.

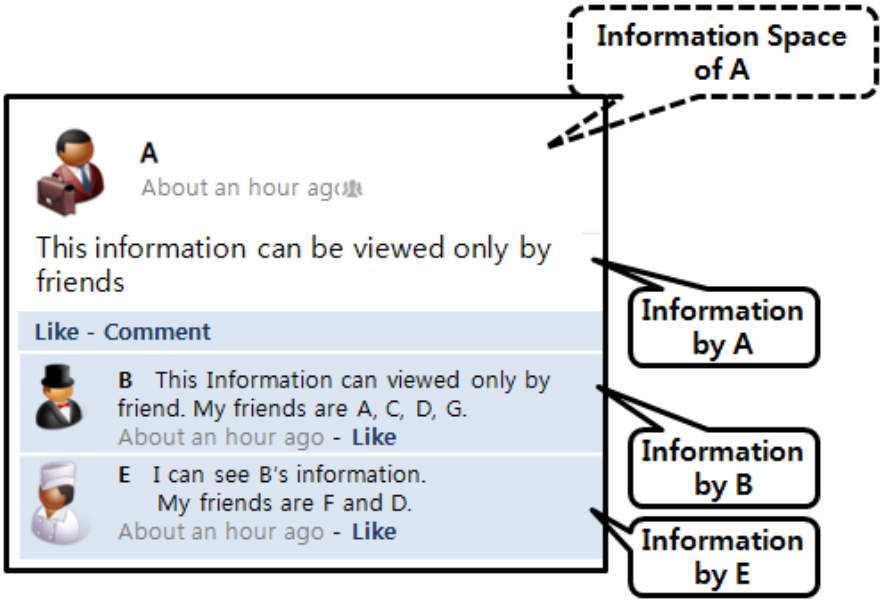

Figure 2. Exposure of Private Information over Facebook

In this paper, we use the following definitions to analyze the flow of information in an SNS [15, 16].

- Information: posts, comments, photos, profiles, *etc.*, *i.e.*, all the information in a user-generated SNS.
- Information Creator: Users who publish information over the SNS.
- Information Space: Data space allocated to a user where user-generated information is stored in the SNS.
- Information Holder: Owner of an Information Space. Information is generated either in the Information Space of the holder or in that of another user. SNS users can publish information. The Information Holder and Information Creator need not be the same.

In Figure 2, the definitions of User B and User E can be used to generate their information in user A's Information Space. Information Creator User B, User E is not have access control. Access control is available only to the owner of the Information Space. Thus, User E can see user B's Information.

SNSs experience security problems such as the sharing of information with other users when the Information Creator does not want to share it with unknown and unconnected users[7, 11].

Figure 2 shows that information leakage can occur over a short distance. However, long-distance information leakage can also occur. Figure 3 shows that User B's information can also be leaked to Users H or F.

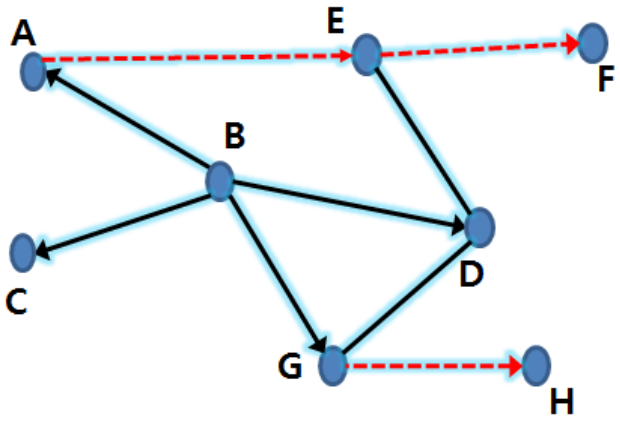

Figure 3. Information Leakage Graph

3. Proposed SNS Access Control Method

In this paper, we propose a hash-chain-based access control information leakage prevention scheme. Traditional SNS services provide access control only to the Information Holder. Thus, access control in traditional SNSs cannot respond to information leakage via authorized users. It also validates the tickets distributed by the Information Holder and the Information Creator. These points provide access control on the sides of the Information Holders and the Information Creator, which prevents information leakage only via unauthorized users

3.1. User Registration Phase

When users join an SNS, they enter their basic information such as their name and e-mail address. The additional information entered is described in Table 2.

Table 2. Additional Information

Additional Information	Scheme	Explanation
Friend Count	Count(n)	A count of the total friend relationships
Master Hash	Mh	The default value used when generating the hash chain
Hash Chain	$h_{count()}(Mh)$	Operation applied to the values of Mh and Count(n)

Additional information is stored when a friend requires access to information, which is used to create and distribute a ticket for verification. A counter is used to calculate whether the user has a friend relationship. Tickets are used to distribute the computation with the stored Mh.

3.2. Friend Relationships

User B sends a friend request to User A. User A requires additional information of User B. User A computes the hash chain of the increased counter value and sends it to User B. User B

receives the hash chain value, which is stored as the value of the ticket. The ticket is used when User B requires access to User A's information

3.3. Generating Information

SNS users generate information, which is divided into the following four categories [14].

- Information generated in their Information Space.
- Information generated in another user's Information Space.
- Comments generated in their Information Space.
- Comments generated in another user's Information Space.

3.4. Ticket Verification

Ticket verification occurs when there is a request for information. A ticket is used to verify whether a legitimate ticket has access control. The verification process checks that the received ticket operations are the same as the Mh and Count(n), which are distributed with their tickets.

3.5. Information Access

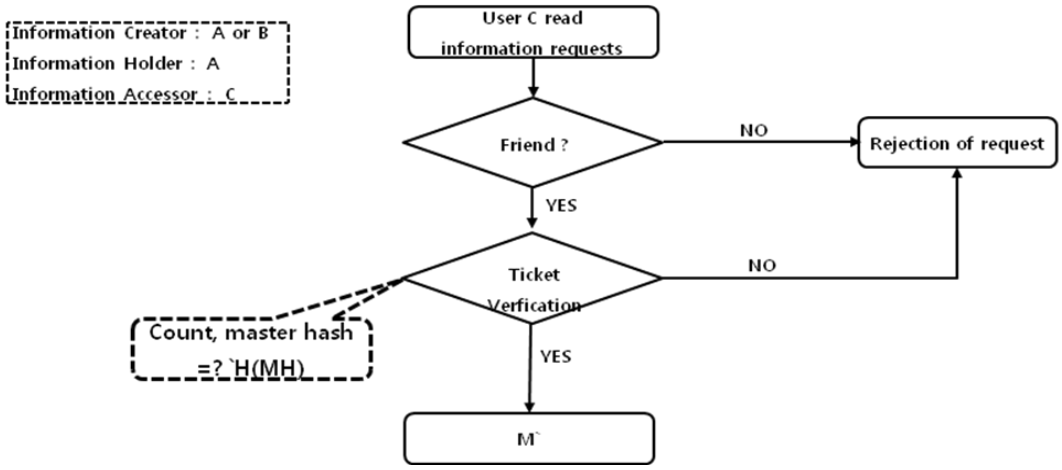

Figure 4. Information Access

1. User C requests that the server give access to User A's Information. The Information Requester, User C, transfers the values of the tickets to the server.
2. The server verifies the relationship.
3. If the Information Requester has a friend relationship, the server validates the ticket value.

3.6. Comment Access

The process allows the reading of comments, and users have access to only the Information Space containing the generated comments that are available upon request. Thus, the steps in Section 3.5 (information access) only address user requests.

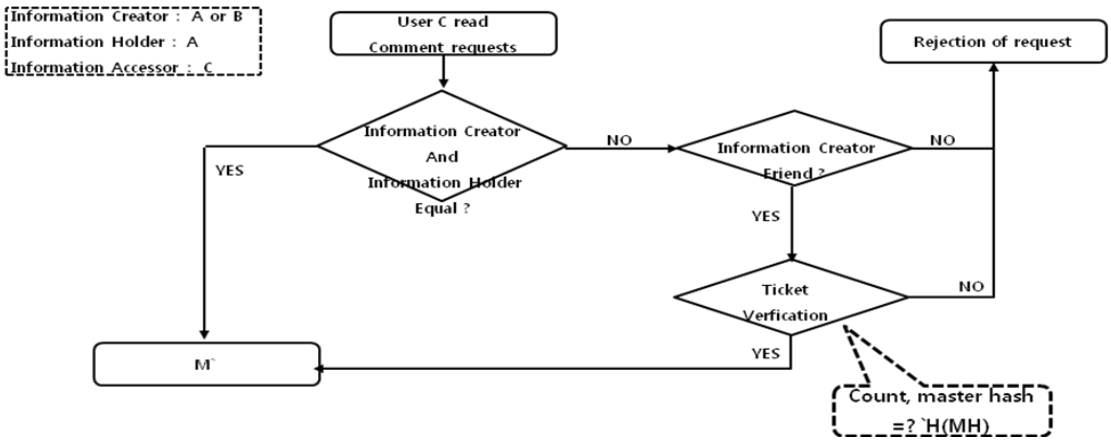

Figure 5. Comment Access

1. User C accesses User A's comments after sending a request to access User A's Information Space via the server. The Information Requester, User C, transfers the values of the tickets to the server.
2. The server checks whether the Information Creator and Information Holders share access. This approach facilitates the validation of the ticket because shared information access is checked for the Information Holders and Information Creator during the phase described in Section 3.5. However, if the information access is not the same, the server completes the verification phase to test the relationship between the requester and the Information Creator.
3. The server verifies the ticket to determine the relationship with the Information Creator.
4. The server validates the ticket value to check that the Information Requester has a friend relationship.

4. Analysis

4.1. Analysis of Additional Space

If the proposed method is applied in the existing SNS environment, additional data such as the Mh value and count(n) of the ticket value are known. A cryptographically secure hash function has a length of 256 bits. The count value expresses the serial number when the friend relationships increase. SNSs have many users, *e.g.*, Facebook has over 1,000 million users, so a user is assumed to have up to 1,000 million friendships, and the count value can increase up to a maximum of 1,000 million. This value is expressed as 16 bits.

Therefore, the application of the proposed method requires additional user information in the space, based on the following formula.

$$\textbf{(256 bit + 16 bit} \times \textbf{256 bit) = 4352bit = 544 Byte}$$

When the proposed method is applied, the additional space is approximately 544 bytes per user.

4.2. Analysis of Additional Time

In this study, we assume that it is safe to use the SHA-256 hash function. The length of information written in the SNS may vary, but 140 B is assumed to be the average.

The hash function in the proposed method is used to distribute tickets to a friend, to validate the ticket of the user who requested the information, and to generate a hash value, which is dumped. This analysis considers the additional time required to request real-time information[5, 6].

When the hash function is used in real-time to request information, One used to read information. And, Two used to read comments. Using a 1.83-GHz Intel Core 2 processor with a Windows Vista 32-bit environment, an SHA-256 operating at 0.275 ms requires.

$$0.550 \times 100 \times 1{,}000 = 55{,}000\ \mu s = 0.055\ s$$

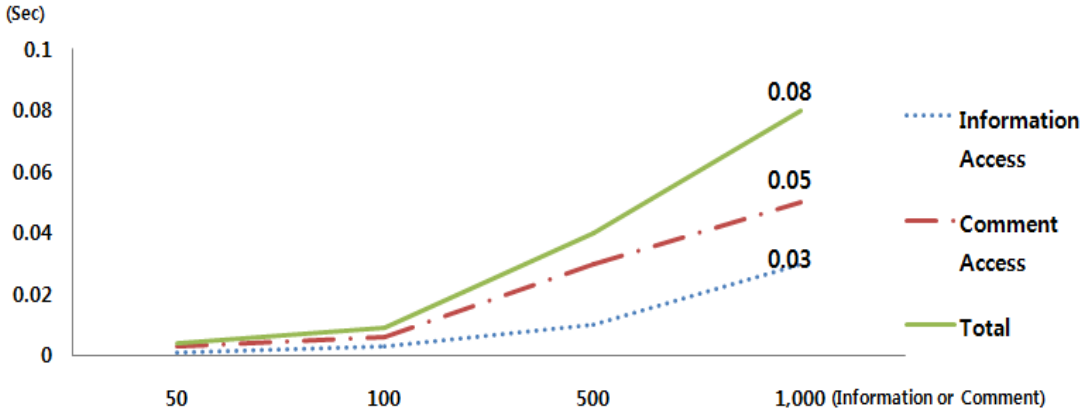

Figure 6. Additional Time Required With Respect to the Number of Users

Figure 6 shows the additional time required with the increasing number of users based on the analysis of this expression

4.3. Security Analysis

Traditional SNS uses access control only Information Space. but, do not provide access control on the Information Creator. SNS users access the information in their group of friends but their information may be exposed to non-friends. Thus, publicly available information may be exploited for malicious purposes. This is known as privacy exposure.

However, in this paper both the Information Space and Information Creator to provide access control. We generated tickets using a cryptographically secure hash function. In addition, an algorithm that considered the Information Creator and Information Holder on both sides was used to validate the ticket. Only users with legitimate tickets were allowed to access information. This prevented information leakage via authorized users.

5. Conclusion

Social networking services (SNSs) allow users to connect via a network. They make it easy to share a variety of information and SNSs have developed rapidly through

information sharing. But, SNS users access the information in their group of friends but their information may be exposed to non-friends, publicly available information may be exploited for malicious purposes. This is known as privacy exposure.

Traditional SNS uses access control only Information Space. But, do not provide access control on the Information Creator.

The approach proposed in this paper generated secure tickets using a cryptographically secure hash function. In addition, the algorithm considered the Information Creator and Information Holder on both sides when validating the ticket. Only users with legitimate tickets were allowed to access information. The Information Holders and Information Creator on both sides of the ticket validation process were configured to control access to different Information Creators and Information Holders. This prevented information leakage via authorized SNS users.

Acknowledgments

This research was funded by the MSIP(Ministry of Science, ICT & Future Planning), Korea in the ICT R&D program 2013.

This work was supported by the Soonchunhyang University Research Fund.

References

[1] L. Young and A. Quan-Haase, "Information Revelation and Internet Privacy Concerns on Social Network Sites: A Case Study of Facebook", Proceedings of the Fourth International Conference on Communities and Technologies, ACM Press, **(2009)**, pp. 265-274.
[2] S. Kim, Y. D. Chung and M. K. Sung, "Data-Centric Access Control Using Relationship-Shell: An Advanced Privacy Model for Social Network Services", 2012 Progress Report for Journal of KIISE, vol. 39, no. 4, **(2012)**, pp. 261-269.
[3] K-J. Kim, S.-P. Hong and J. Young Kim, "A Study on Policy-based Access Control Model in SNS", IJMUE, vol. 7, no. 3, **(2012)**, pp. 143-150.
[4] M. Meng, N. Zakaria, S. Bindahman, N. Mohd Asrol Alias and W. Husain, "RivacyDoc: A Study on Privacy Protection Tools for Children in SNS", ISHE, vol. 3, no. 6, **(2012)**, pp. 41-48.
[5] Speed Comparison of Popular Crypto Algorithms, available http://cryptopp.com/benchmarks.html.
[6] H. Jeong and D. Won, "A Method for the Protection of the Message in Social Network Service (SNS) using Access Control and Hash-chain", 2013 Progress Report for Journal of KIISC, vol. 1, no. 23, **(2012)**, pp. 81-88.
[7] R. Gross and A. Acquisti, "Information Revelation and Privacy in Online Social Networks", Proceedings of the 2005 ACM workshop on Privacy in the Electronic Society, **(2005)**, pp. 71-80.
[8] J. Koo, "The right to informational self-determination of internet users", Informatization policy, vol. 10, no. 3, **(2003)**.
[9] D. M. Boyd and N. B. Ellison, "Social network sites: definition, history, and scholarship", Journal of Computer Mediated Communication, vol. 13, no. 1, **(2007)**.
[10] S. Wasserman and K. Faust, "Social network analysis: methods and applications", Cambridge University Press, **(1994)**, pp. 71-76.
[11] L. Sweeney, "k-anonymity: A model for protecting privacy", International Journal on Uncertainty, Fuzziness and Knowledge-based System, vol. 10, no. 3, **(2002)**, pp. 557-570.
[12] A. Machanavajjhala, D. Kifer, J. Gehrke and M. Venkitasubramaniam, "L-diversity: Privacy beyond k-anonymity", Proceedings of International Conference on Data Engineering, **(2006)**, pp. 24.
[13] N. Li, T. Li and S. Venkatasubramanian, "tCloseness: Privacy Beyond k-Anonymity and lDiversity", Journal ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 1, no. 1, **(2007)**.
[14] X. Xiao and Y. Tao, "M-invariance: towards privacy preserving re-publication of dynamic datasets", Proceedings of the 2007 ACM SIGMOD international conference on Management of data, **(2007)** June 11-14.
[15] B. Carminati, E. Ferrari and A. Perego, "Rulebased access control for social networks", OTM Workshops, vol. 2, **(2006)**, pp. 1734-1744.

[16] B. Carminati and E. Ferrari, "Privacy-Aware Collaborative Access Control in Web-Based Social Networks", Proceeedings of the 22nd annual IFIP WG 11.3 working conference on Data and Applications Security, **(2008)** July 13-16.

Authors

Yu-Jong Jang received his B.S. degree in Information Security from Soonchunhyang University (SCH), South Korea, in 2012. He is currently an M.S. candidate in the Information Security Application and Assurance Lab at Soochunhyang University. His research interests include cryptographic protocol, protection of privacy in SNSs, and cloud computing security.

Jin Kwak received his B.S. (2000), M.S. (2003), and Ph.D. (2006) from Sungkyunkwan University (SKKU) in Korea. Prior to joining the faculty at Soonchunhyang University (SCH) in 2007, He joined Kyushu University in Japan as a visiting scholar. After that, he served MIC (Ministry of Information and Communication, Korea) as a Deputy Director. Also, he have served as a Dean of DISE(2009-2010) and Vice-Dean of College of Engineering (2009) in SCH. Now he is a Professor of Department of Information Security Engineering (DISE) at SCH. His main research areas are cryptology, information security applications and information assurance.