Poster: Knowledge-Based Authentication using Twitter

Tomofumi Nemoto
Kanagawa Institute of Technology
Shimo-ogino 1030 Atsugi-shi
Kanagawa, Japan
nemohi@gmail.com

Kyohei Furukawa
Kanagawa Institute of Technology
Shimo-ogino 1030 Atsugi-shi
Kanagawa, Japan
nullpo.g@gmail.com

Manabu Okamoto
Kanagawa Institute of Technology
Shimo-ogino 1030 Atsugi-shi
Kanagawa, Japan
manabu@nw.kanagawa-it.ac.jp

1. INTRODUCTION

Knowledge-based authentication (KBA) [1]-[4] is a method of authentication that verifies the identity of a person accessing a service such as a website. In a KBA scheme, the user is asked to answer a secret question for authentication. KBA is often used as multifactor authentication or for self-service password retrieval.

Secret questions can be static or dynamic. In a static scheme, the user needs to select the question(s) he would like to be asked and provides the answer(s) to it(them) before he can actually use the site. The question/answer pairs are stored in the service provider and are used to verify the user's identity when he connects to the system. In a dynamic scheme, however, how we store question/answer pairs is a problem.

In this paper, we describe the use of Twitter to store question/answer pairs. In this scheme, users sometimes answer the question from the service provider as a Twitter dynamic message and the question/answer is stored.

2. WHY KBA?

Almost all websites used ID/passwords as a means of user authentication. Such ID/password schemes, however, are problematic. Users may forget their password. Users may write down their passwords to remember them, but this written reminder may then be stolen or lost. In addition, users typically use many websites and must therefore remember many ID/passwords for these different sites.

KBA is a method that solves these password problems. KBA asks the user an easy and factual question that the user does not need to remember. Some examples of KBA questions are: "What city were you born in?" and "What color is your car?" The questions may also be about preferences, such as: "What is your favorite food?" or "Who was your favorite pet?" KBA requires multiple questions/answers for a single authentication because someone who is not the true user can conceivably provide a common correct answer, such as “New York” or “red car.”

In a static scheme, an ill-intentioned user can intercept or spy upon the message that contains the questions/answers exchange between the user and the KBA system, and then record the question/answer pairs and use them to provide authentication. For this reason, users should not use same questions and answers for a long time. This is why KBA should be used in a dynamic scheme.

In a dynamic scheme, we need to regularly and even frequently obtain new question/answer pairs. Requiring users to enter question/answer pairs when they connect to a website, though, is then burdensome. What is needed is a way to obtain that information automatically. For example, [5] is a method that employs e-mail to store personal information for KBA.

In this paper, we use Twitter [6] to store personal information for KBA. Twitter is a social networking and microblogging website that that enables its users to send and read messages called tweets. Tweets are text-based posts of up to 140 characters that are displayed on the user's profile page. This approach is more convenient and user-friendly than e-mail. Twitter can be accessed with either a cell-phone or palmtop computer.

3. PROPOSED METHOD

In this section we propose a KBA scheme that uses Twitter. We describe the method using a simple example.

The website that the user wants to access is “Service Provider” (SP). SP uses KBA for user authentication. SP needs to store the user’s question/answer pairs.

The user needs to have an account on Twitter in order to use Twitter for KBA. (The user can of course use Twitter as usual as a microblog.)

The Service Provider also needs to have a Twitter account in order to obtain the user’s information for KBA. The SP and user need to follow each other’s accounts. This enables both the SP and user can send direct messages to each other on Twitter. Figure1 presents a schematic of this system.

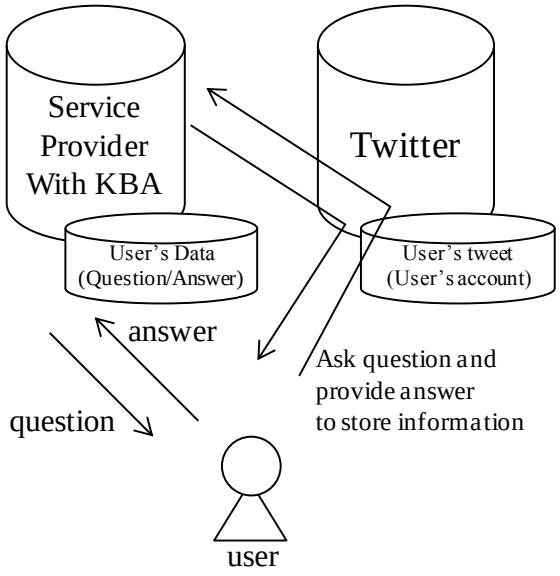

Figure 1. KBA system with twitter

Here, we describe the steps in our proposed KBA scheme.

1) SP sends a direct message on Twitter to the user Alice. This message is a simple question such as "What are you having for lunch today?"

2) Alice replies to this direct message on Twitter by sending an answer such as "Cheeseburger." Any other users or followers on Twitter cannot read this question/answer pair because they use "direct message" on Twitter.

3) SP stores question-answer-user pairs. In this case, it is: "Lunch on February 15" – "Cheeseburger" – "Alice." SP repeats step 1-3 and can use these pairs for KBA.

4) Alice wants to access SP. SP needs to authenticate Alice by KBA. Alice sends her ID to SP.

5) SP asks a question based upon information stored in Step 3, above: Thus: "What did you have for lunch on February 15?" Alice answers this question at once. SP needs to set a brief time period in which Alice can answer the question, because others may search for that information in various ways in order to impersonate Alice.

6) Alice answers the question and SP confirms the answer to authenticate Alice. SP repeats these steps several times. Alice has to answer all questions correctly.

7) SP authenticates Alice and she can then use any services on SP.

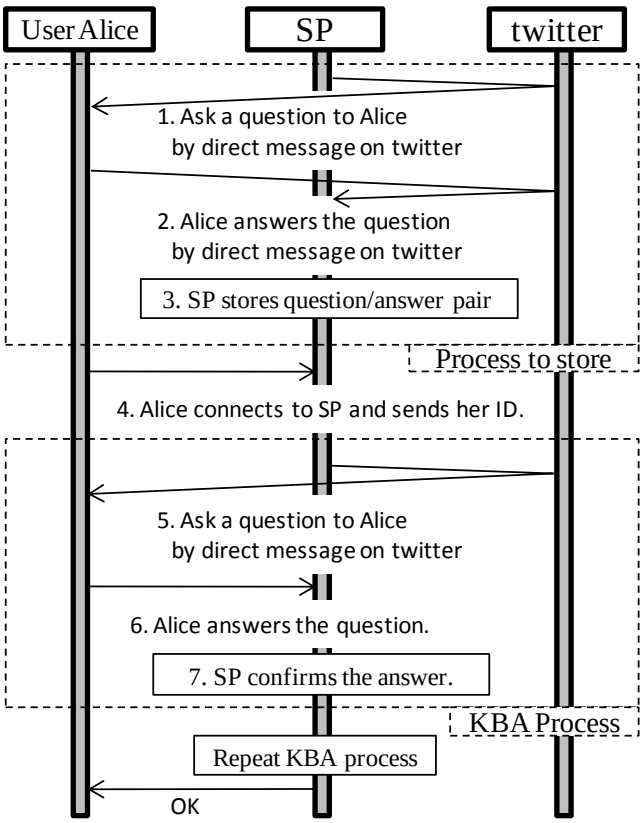

Figure 2. KBA sequence

In Step 5, SP can also ask a question via a direct message on Twitter. At that point, in Step 3, SP says: "We've sent a question to your Twitter account. Check there and enter you answer here" to Alice on the website. Alice checks the direct message on Twitter. Messages on Twitter are questions for KBA. Alice, though, answers the question not on Twitter but on the website. By adopting this new step, KBA becomes more secure because no one can even read the questions and SP can also confirm whether the user is the same as the holder of the Twitter account.

Figure 2 shows the sequence of this system.

4. FUTURE WORK

In this scheme, a question must be set in SP prior to authentication, and the question must be static. We need to set a variety of questions in SP that address other topics, such as lunch, dinner, tie color, etc.

We can store these questions in SP using Twitter. To do so, the user may send a direct message to SP with a special tag such as "#kba."

When SP receives the direct message with the tag #kba, SP analyzes the contents of the message and obtains both the question and answer in the message along with the date. For example, for the direct message: "Today's dinner is Japanese sushi. That's good. #kba", SP can store such a question-answer-user pair as "Dinner on May 22" – "Sushi" – "Bob." In this scheme, however, users must send their messages to SP frequently.

5. CONCLUSION

In this paper, we proposed a knowledge-based authentication scheme that uses Twitter, which enables us to obtain question/answer pairs easily. Users only need to answer the question, which can sometimes be sent by direct message on Twitter, and SP can store the user's information for KBA.

6. REFERENCES

[1] KBA (Knowledge Based Authentication), http://csrc.nist.gov/archive/kba/.

[2] Muhammad Daniel Hafiz Abdullah, Abdul Hanan Abdullah, Norafida Ithnin, Hazinah Kutty Mammi, "Towards Identifying Usability and Security Features of Graphical Password in Knowledge Based Authentication Technique," Asia International Conference on Modelling & Simulation, pp. 396-403, 2008 Second Asia International Conference on Modelling & Simulation, 2008.

[3] Ye Chen, Divakaran Liginlal, "A maximum entropy approach to feature selection in knowledge-based authentication," Decision Support Systems, Volume 46, Issue 1, pp.388-398, December 2008.

[4] Rachna Dhamija and Adrian Perrig, "a user study using images for authentication," In Proceedings of the 9th conference on USENIX Security Symposium, Vol.9, 4-4, 2000.

[5] M.Nishigaki, M.Koike, "A User Authentication Based On Personal History: A User Authentication System Using E-mail History," IPSJ Journal, Vol.47(3), pp.945-956, March 2006.

[6] Twitter Inc, http:// twitter.com/.