On Authentication between Human and Computer

Takahiro Watanabe Yasunobu Nohara Kensuke Baba Sozo Inoue
Hiroto Yasuura
Kyushu University
6-1 Kasugakoen, Kasuga-City, Fukuoka 816-8580, JAPAN
{takahiro,nohara,baba,sozo,yasuura}@c.csce.kyushu-u.ac.jp

Abstract

Electronic authentication with a portable device such as a smart card has been receiving increasing attention. In an authentication, the portable device is regarded as the human user himself. However, in an open environment like authentication systems, it is necessary to have a way of secure communication between the portable device and the human user. This paper considers an authentication of a server computer of a service provider by a human with a portable device as a part of the authentication and an attack by a client computer which relays the communication between the portable device and the server computer. As a defense against the attack, we introduce a system with a portable device which has an interface to show information to a human.

1 Introduction

In recent years, electronic authentication has been receiving increasing attention by the explosive spread of computer networks. In this paper, our particular focus is the authentication between a human user and a server computer.

Papers by Hopper and Blum[1], Matsumoto and Imai [2] provide schemes which authenticate a human by a computer. However, the scheme which authenticate a server computer by a human is necessary to prevent cheating by the counterfeit service which is becoming a crucial problem as the way of phishing.

One of the mutual authentication methods is human using a token such as a smart card. In an authentication between a token and a server computer, it is likely that various cryptographic technologies realize a secure mutual authentication where a server computer is authenticated by the human user and vice versa. A token which has sufficient computation ability can put into practice such a secure authentication even on a untruthful network [3, 4, 5, 6].

However, when a human authenticate the computer by a token, we should not consider security only between the token and the server. First, we should consider situation where the token is stolen. If the human user has the token stolen, the server computer might authenticate the illegitimate human user who stole the token from the human user. Biometrics is, in a sense, a solution to fill the gap between a human and a token. This technology guarantees the correspondence between a token and its owner in a human authentication by a server computer.

Moreover, we should consider the communication between the human and the token. If we use a token which does not have an interface to show information to the human user(such as the smart card), the result of the authentication by a token might be relayed to a client computer. For example, we consider an attack from the counterfeit ATM (Automatic Teller Machine) (Fig.1).

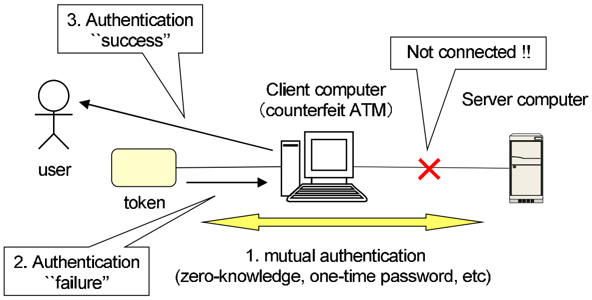


Figure 1. An attack by a untruthful client computer

The following procedure is the attack by the ATM:

1. The token authenticates the server computer.

2. The token sends the authentication result which is "failure" to the client computer.

Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)
0-7695-2520-2/06 $20.00 © 2006 IEEE

3. The client computer displays the authentication result which is "success".

By this procedure, even if a token correctly authenticate the server computer, the client computer can cheat the human.

When human users interact with services through a client computer, the authentication is often done by a human user providing a token like a smart card. This establishes an authenticated link between the token and the server computer. However, what our society requires is a secure authentication between a "human user" and a server computer, and it is not realized straightforwardly from a secure authentication between a "token" and a server computer. This argument is important since some technologies for the secure authentication are constructed under the assumption that the client computer is trusted.

In this paper, we consider a model of an authentication between a human and a server computer with a token and a client computer, especially the authentication of the server computer by the human. This model divides the human and the token explicitly, which makes clear the essence of the problem of an untruthful client computer. Then, as a solution to this problem, we introduce an authentication protocols with a portable device which has an interface to show information to a human.

2 Modeling An Authentication Systems

The target of this paper is an authentication of a server computer by a human. It is often confused with the authentication by a token, hence we consider the following model system which divides a human and a token explicitly.

The authentication model is constructed by the following four objects:

- $server\{s\} \in \{s_r, s_f\}$: is a server computer of a service provider and wishes to authenticate a human. s_r is a real server, and s_f is a fake server.
- $user\{u_i\}(1 \leq i \leq N)$: is a human user who wishes to have a service from a service provider and has a $token$.
- $token\{t_i\}(1 \leq i \leq N)$: is a portable device which has a computation ability for a secure mutual authentication with other computers, and is trusted by a user.
- $clientcomputer\ c$: is a computer which relays the communication between a service computer and a $token$ and has an interface to give information to a human.

Then, we assume that a token and a server can operate a secure mutual authentication, that is, they send information for the authentication each other and the information gives no knowledge to the other objects. Moreover, a client computer can know any information sent between a token and a server, however the token or the server can know whether the client computer tampered the information.

Now we formalize the communications between objects by following three relations:

- a server can send information to a client computer.
- a token can send information to a client computer.
- a client computer can send information to a server, a user, or a token.

These relation is illustrated as in Fig. 2.

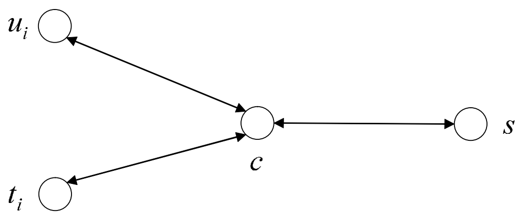


Figure 2. The relations between the objects

Under the previous relations, (a trial of) a mutual authentication between a user ($u_i \in \{u_i\}$) and a server ($s \in \{s_r, s_f\}$) is operated by following procedure (Fig. 3):

1. the server and the token of the user operate a mutual authentication.
2. the token sends the result $r_{tc} \in \{1, 0\}$ of the authentication to the client computer. Where 1 and 0 correspond to information that the server was accepted/rejected by the token. .
3. the client computer shows the result $r_{cu} = r_{tc}$ to the user,

By the assumption, a mutual authentication between the token and the server is operated securely during the first step. If the server can confirm the correspondence between the token and its owner (for example, by a technology of biometrics), the server can identify the user. On the other hand, as to the authentication of the server by the user, the user can not confirm whether $r_{tc} = r_{cu}$. Therefore, the client computer can success the attack to connect with a fake server besides tampering the information.

In this paper, we consider an authentication of a server by a user with a trusted token and an attack by a client computer against the authentication. A protocol of an authentication of a server by a user is *valid* if the user can know that the server is rejected (even if the user can not know that the

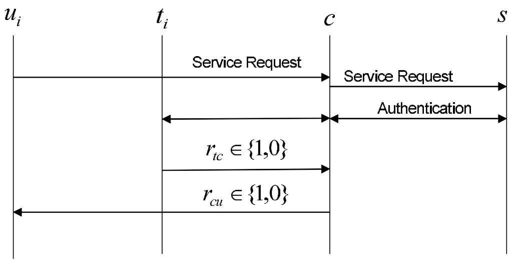


Figure 3. An authentication protocol between a user and a server computer

server is accepted). It is clear that no protocol can be valid on the previous model since the client computer can send $r_{cu} = 1$ for $r_{tc} = 0$ if the client computer is an attacker.

To prevent the attack from client computer, the authentication protocol between the user and the server computer should satisfy either Condition 1 or Condition 2.

Condition 1 The client computer can not send "$r_{cu} = accept$" if the client computer receive "$r_{tc} = reject$".

Condition 2 The user can confirm "$r_{tc} = r_{cu}$".

3 The Token With Interface To The User

We introduce a model of an authentication system which realizes valid protocol of the authentication of a server by a human. This model is essentially an extension with respect to the communication of the objects.

The problem on the model in the previous section is that the user can know the result of the authentication between the token and the server only by the information from the client computer. A straightforward solution is to consider a protocol of an authentication of the client computer by the user which guarantees the information from the client computer. This solution is realized, for example, by a password (as a challenge of an authentication of the client computer) of user. Another solution is to establish a connection from a trusted object to the user for the result of the authentication. In this section, we consider a model which has a connection from the token to the user, that is, we add to the communication in the previous section the following condition:

- a token can send information to a user.

This relation is illustrated as in Fig. 4.

In the model of the previous connection, a mutual authentication between a user $(u_i \in \{u_i\})$ and a server $(s \in \{s_r, s_f\})$ is operated as follows (Fig. 5):

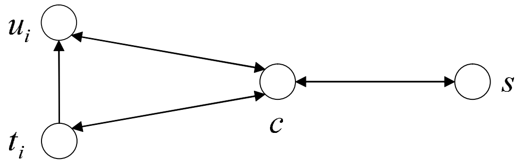


Figure 4. The relations between the objects (The token with the interface to a user)

1. the server and the token of the user operate a mutual authentication
2. the token shows the result $r_{tu} \in \{1, 0\}$ of the authentication to the user. Where 1 and 0 correspond to information that the server was accepted/rejected by the token.

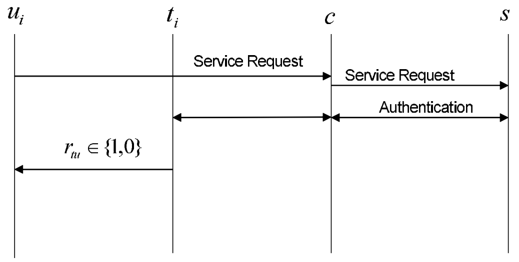


Figure 5. An authentication Protocol (Token with the interface to a user)

By the assumption that the token is trusted, r_{tu} is equal to the result of the authentication. Therefore, the protocol of the server authentication is valid.

4 Discussion

We introduced a model of an authentication between a human and a server computer with a trusted token and a client computer. In the model, we argued about an authentication of the server computer by the human and an attack by the client computer. The attack of the client computer can be prevented if one of the following conditions is satisfied.

Condition 1 The client computer can not send "$r_{cu} = accept$" if the client computer receive "$r_{tc} = reject$"

Condition 2 The user can confirm "$r_{tc} = r_{cu}$".

If the token for an authentication has an interface to a human, Condition 2 is satisfied and therefore the attack is prevented. Thus, we consider that a portable device with a suitable interface (for example, a smart card with an LED) is one of the solutions to improve a kind of security on electronic authentication in our daily life.

In practice, a large number of portable devices with no interface have been already spread. Thus, our future works will focus on other methods to guarantee the conditions. It seems that, on the introduced model, the user can not confirm the result without any connection with the token. we consider that one of the practical solutions is to use an additional portable device such as a cell phone which can receive the result from the server. The user can know the result by using the cell phone when the user want to confirm the result. But it may be expensive that the user always use the cell phone to authentication. Thus, this method might not be able to be used for the application which uses the un-trusted client computer many time.

In the rest of this paper, we describe the outline of an idea to realize the condition on the introduced model.@ We add a following object in previous model.

- $name\{a_i\}(1 \leq i \leq N)$: is the name of user $u_i \in \{u_i\}$ and stored in the token $t_i \in \{t_i\}$.

Then, the authentication protocol between a user $(u_i \in \{u_i\})$ and a server $(s \in \{s_r, s_f\})$ is operated (Fig.6):

1. the server and the token of the user operate a mutual authentication.
2. the token sends the result. If it is $accept$, then the token send the client computer $r_{tc} = a_i$. Otherwise, the token sends $r_{tc} = 0$.
3. the client computer shows the result $r_{cu} = r_{tc}$ to the user,

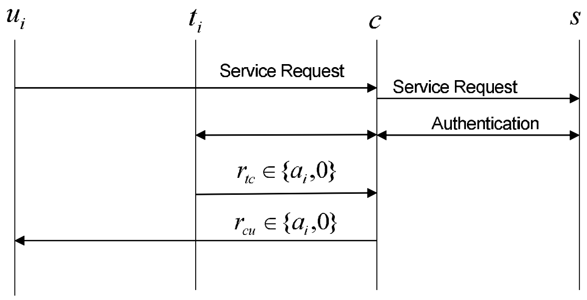


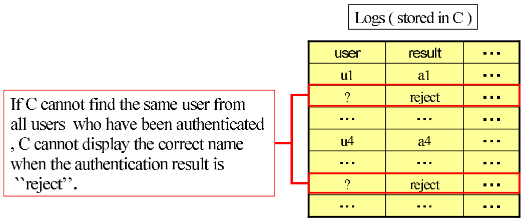
Figure 6. An authentication Protocol (Token sends a user name)

In this system, the user (u_i) judges that the authentication is success if the client computer displays the name (a_i). Thus, when the token sends 0 to the client computer and the client computer displays the name, the user is cheated .

The client computer which is able to communicate with real server can know the user's name. But if the client computer cannot find the user from all users who have been authenticated in the past, the client computer cannot display the correct user name a_i.

The property that a user may make multiple uses of services without others being able to link these uses together is called *unlinkability* [7].

Thus, if this authentication system can protect the unlincability between users against the client computer, the user is not cheated.



Logs (stored in C)

user	result	···
u1	a1	···
?	reject	···
···	···	···
u4	a4	···
···	···	···
?	reject	···
···	···	···

If C cannot find the same user from all users who have been authenticated , C cannot display the correct name when the authentication result is ``reject''.

Figure 7. Unlinkability between users

References

[1] Hopper, N.J. and Blum, M., "Secure Human Identification Protocols", ASIACRYPT 2001, LNCS, vol.2248, pp.52–66, 2001.

[2] Matsumoto, T. and Imai, H., "Human Identification Through Insecure Channel", EUROCRYPT 91, LNCS, vol.547, pp.409–421, 1991.

[3] Fiat, A. and Shamir, A., "How to prove yourself: Practical solutions to identification and signature problems", CRYPTO '86, LNCS, vol.263, pp.186–194, 1987.

[4] Lamport, L., "Password authentication with insecure communication", Communications of the ACM, vol.24, no.11, pp.770–772, 1981.

[5] Itoi, N. and Honeyman, P., "Smartcard integration with Kerveros V5", USENIX Workshop on Smartcard Technology, Chicago, May 1999.

[6] Aviel D. Rubin., "Independent one-time passwords", USENIX Journal of Computer Systems, February 1996.

[7] "ISO/IEC 15408 - INTERNATIONAL STANDARD Information technology - Security techniques - Evaluation criteria for IT security - Part2: Security functional requirements", Dec.1999.