International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:4, No:3, 2010

A Proposal for Federation Technology for Authenticated Information between Terminals

Katsuyuki Umezawa, Takashi Tashiro, and Satoru Tezuka

***Abstract*—**Recently, various services such as television and the Internet have come to be received through various terminals. However, we could gain greater convenience by receiving these services through cellular phone terminals when we go out and then continuing to receive the same services through a large screen digital television after we have come home. However, it is necessary to go through the same authentication processing again when using TVs after we have come home. In this study, we have developed an authentication method that enables users to switch terminals in environments in which the user receives service from a server through a terminal. Specifically, the method simplifies the authentication of the server side when switching from one terminal to another terminal by using previously authenticated information.

***Keywords*—**Authentication, Cookie, Federation, GBA, IPSec

I. Introduction

RECENTLY, a lot of people have come to use cellular phone terminals to receive various services such as television, the Internet, and so on. Moreover, a variety of electronic equipment has come to be connected continuously with high-speed broadband lines. This means that greater convenience is possible if services are received through the cellular phone terminals when we go out and the same services are then received through large screen digital televisions after we have come home. If authenticating digital TVs after authenticating our cellular phones becomes unnecessary, services will be received even more conveniently.

In this study, we have developed an authentication method that simplifies the authentication processing between terminal 2 and the server by coordinating information that has been authenticated between terminal 1 (that has been used before) and terminal 2 (that will be used after).

In Section 2, we describe three authentication methods (cookies, generic bootstrapping architecture (GBA), and IPSec) that are relevant to this work. In Section 3, we describe the proposed method. Specifically, we detail how the method coordinates information that has been authenticated in order to use the abovementioned three authentication methods. In Section 4, we describe our implementation using Cookie method, and evaluate them in Section 5. Section 6 is the conclusion.

Katsuyuki Umezawa and Takashi Tashiro are with the Hitachi Ltd., Systems Development Laboratory, 292, Yoshida-cho, Totsuka-ku, Yokohama-shi, Kanagawa, 244-0817 JAPAN.

Satoru Tezuka is with the School of Computer Science, Tokyo University of Technology 1404–1, Katakuramachi, Hachioji City, Tokyo 192–0982 JAPAN.

II. Related Methods

A. Cookie method

Cookies are small pieces of data text stored by Web browsers. Their principal aim is recycling information. For instance, cookies on a user's visits to a Web site and login information are preserved in his or her Web browser. We can omit updates to visit histories and login processing by transmitting preserved cookie information to a Web site when we access it again. Cookies are provided by RFC2109 [1] and RFC2965 [2]. The basic protocol flow of the cookie method is shown in Figure 1 and explained in Table 1.

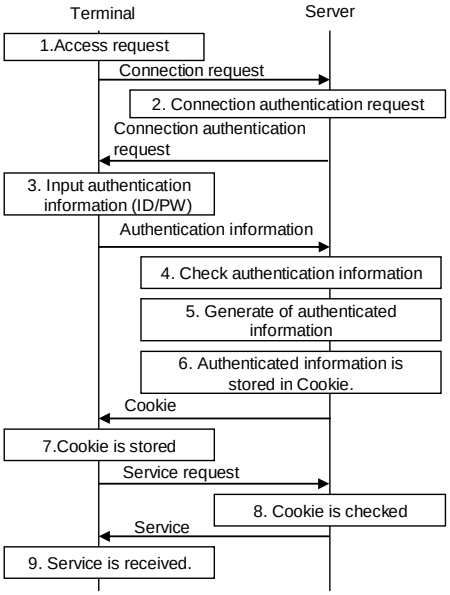

Fig. 1 Basic protocol flow of Cookie method

TABLE I
Explanation of Figure 1

No.	Explanation
1	The terminal sends the connection request to the server.
2	The server sends the connection authentication request to the terminal.
3	The user inputs authentication information (ID, password, etc.) to the terminal. The terminal then sends the authentication information to the server.
4	The server checks whether the authentication information registered in the server corresponds to that sent by the terminal.
5	If the authentication information corresponds, the server generates previously authenticated information.

6	The server stores the authenticated information that has been generated in a cookie and sends it to the terminal.
7	The terminal preserves the cookie. The terminal resends a service request and a cookie to the server at the reconnection.
8	The server checks whether the authenticated information stored in the cookie corresponds to the authenticated information that the server has generated. If it does, the service is provided.
9	The terminal receives the service.

B. GBA method

GBA is an authentication method provided by 3GPP [3] for a mobile operator to authenticate the subscriber. Figure 2 shows the basic system configuration of the GBA method.

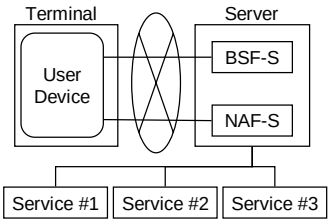

Fig. 2 Basic system configuration of the GBA method

BSF-S refers to the bootstrapping server function server. The BSF-S executes the bootstrap authentication and generates the service key. NAF-S refers to the network application function server. NAF-S provides services for cellular phone terminals.

(1) Bootstrap authentication

Figure 3 shows and Table 2 explains the processing flow of the bootstrap authentication in the GBA method.

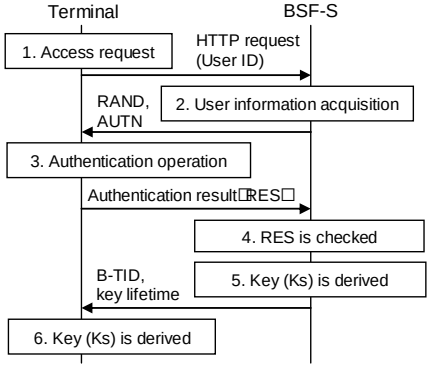

Fig. 3 Basic protocol flow of bootstrap authentication

TABLE II
EXPLANATION OF FIGURE 3

No.	Explanation
1	The terminal sends the user ID to the BSF-S.
2	The BSF-S acquires subscriber information and then transmits random numbers (RAND) and information necessary for authentication operation (AUTN) to the terminal.
3	The terminal checks the AUTN, and calculates CK, IK (Ks = CK‖IK), and RES. The terminal sends the authentication operation result (RES) to the BSF-S.
4	The BSF-S checks the RES.
5	The BSF-S derives a key (Ks). The BSF-S sends B-TID and the lifetime of Ks to the terminal.
6	The terminal generates Ks_NAF by using Ks. The terminal preserves B-TID and Ks until they expire or are updated.

(2) Service authentication

Service authentication is needed in order to receive the service after the abovementioned bootstrap processing has finished authenticating the user. The service key is derived in the service authentication processing. The processing flow of the service attestation is shown in Figure 4 and explained in Table 3.

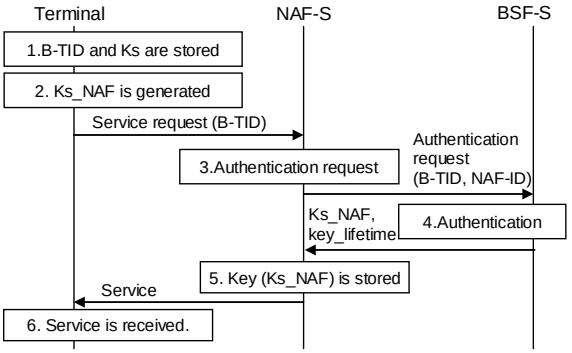

Fig. 4 Basic protocol flow of service authentication processing

TABLE III
EXPLANATION OF FIGURE 4

No.	Explanation
1	As a result of the bootstrap authentication, the terminal stores B-TID and Ks.
2	The terminal generates the service key (Ks_NAF). The terminal requests service from the NAF-S. At that time, B-TID is notified.
3	The NAF-S sends authentication request to the BSF-S, and, at the same time, B-TID and NAF-ID are notified.
4	The BSF-S derives Ks_NAF and sends it to NAF-S.
5	The NAF-S stores Ks_NAF. NAF-S then provides the service to the terminal by using Ks_NAF.
6	The terminal receives the service.

C. IPSec method

This method[4-14] communicates by using information to secure the traffic, security association (SA) in IPSec. There are two kinds of SAs: One that transmits data by using IPSec (IPSec_SA), and one used to generate IPSec_SA (IKE_SA). IKE_SA is generated in unsecure environments. The basic protocol flow of generating IKE_SA is shown in Figure 5 and explained in Table 4.

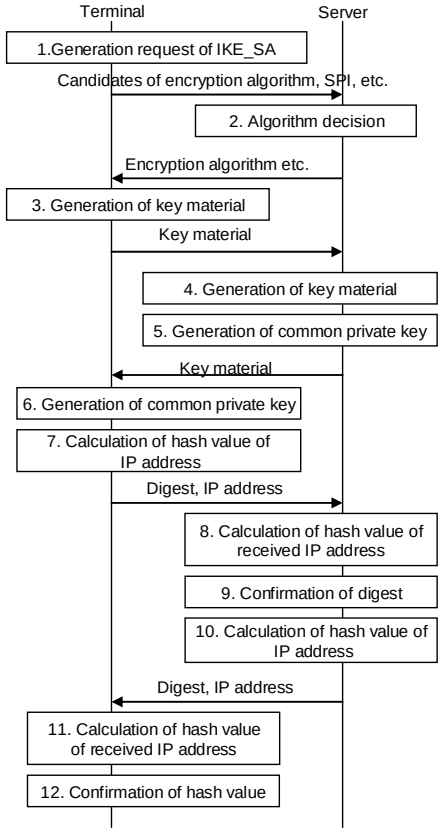

Fig. 5 Basic protocol flow of generating IKE_SA

TABLE IV
EXPLANATION OF FIGURE 5

No.	Explanation
1	The terminal sends the candidates of the security parameter index (SPI), encryption algorithm, pseudo-random number function, completeness algorithm, and Diffie-Hellman group to the server.
2	The server decides the best candidates to use. The server sends the terminal the selected encryption algorithm, pseudo-random number function, completeness algorithm, and Diffie-Hellman group.
3	The terminal generates some of the key material to make a common private key and sends it to the server.
4	The server generates the rest of the key material to make the common private key.
5	The server generates the common private key from the key materials from the server and the terminal. The server sends the terminal the key material of the server.
6	The terminal generates the common private key from the key materials from the terminal and the server.
7	The terminal calculates the hash value of its IP address by using its common private key. The hash value is transmitted to the server.
8	The server calculates the hash value of the IP address of terminal by using its common private key.
9	The server confirms whether the hash value sent in Step 7 corresponds to the hash value calculated in Step 8.
10	The server calculates the hash value of its IP address by using its common private key. The hash value is transmitted to the terminal.
11	The terminal calculates the hash value of the IP address of server by using its common private key.
12	The terminal confirms whether the hash value sent in Step 10 corresponds to the hash value calculated in Step 11.

The generation of IKE_SA is completed in the above way. Next, IPSec_SA is generated by using IKE_SA made in accordance with the abovementioned procedure. The basic protocol flow of generating IPsec_SA is shown in Figure 6 and explained in Table 5.

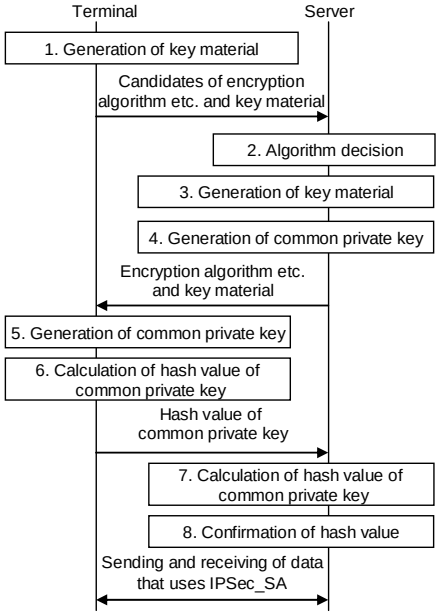

Fig. 6 Basic protocol flow of generating IPSec_SA

TABLE V
EXPLANATION OF FIGURE 6

No.	Explanation
1	The terminal generates the key material used with IPSec_SA. The terminal sends the server the candidates of the SPI, encryption algorithm, pseudo-random number function, completeness algorithm, key material of terminal, range of IP address and range of ports on the terminal side where IPSec is applied, and range of IP address and range of ports on server side.
2	The server decides the candidate that it is most possible to correspond to from among those sent from the terminal.
3	The server generates the key material used with IPSec_SA.
4	The server generates the common private key from the key materials of the terminal and the server. The server sends the algorithm decided at Step 2 and its key material to the terminal.
5	The terminal generates the common private key from the key materials of the terminal and the server.
6	The terminal calculates the hash value of the common private key and sends it to the server.
7	The server calculates the hash value of the common private key.
8	The server confirms whether the hash value received at Step 6 corresponds to the hash value calculated at Step 7 are corresponding.

The generation of IPSec_SA is completed in the above way. Data is sent and received by using this IPSec_SA. The content of the encryption method of IKE_SA and IPSec_SA generated is stored in the security association database (SAD) and the security policy database (SPD).

III. Proposed Method

We have developed an authentication method that simplifies the authentication processing between terminal 2 and the server by coordinating information that has been authenticated between terminals 1 and 2.

Specifically, when switching from terminal 1 to terminal 2, information that has previously been authenticated is made to cooperate between both terminals. Afterwards, terminal 2 receives the service from the server by using the information that has been authenticated.

A. Outline of proposed method

First of all, we describe how terminal 1 is processed in the proposed method. Figure 7 shows and Table 6 explains the general flow when the center authenticates terminal 1 and service is provided.

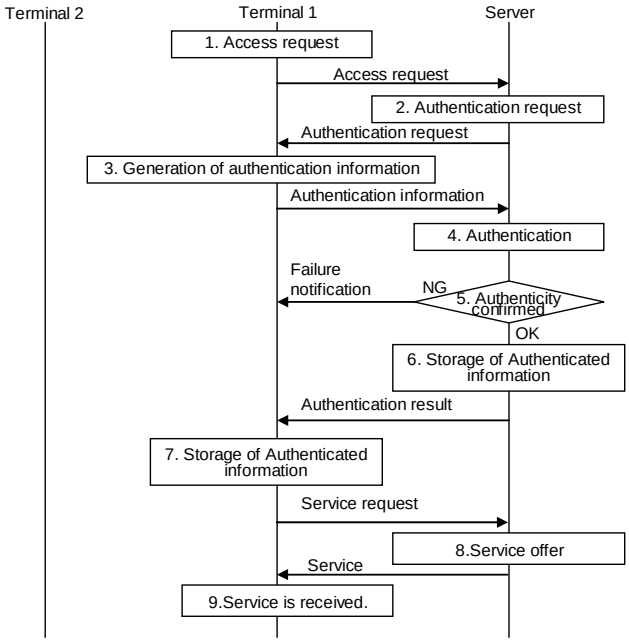

Fig. 7 General flow when the center authenticates terminal 1 and service is provided

TABLE VI
Explanation of Figure 7

No.	Explanation
1	Terminal 1 sends the connection request to the server.
2	The server returns the authentication request to terminal 1.
3	Terminal 1 generates the authentication information and send it to the server.
4	The server processes the authentication.
5	The server notifies the terminal if the authentication processing fails.
6	The server keeps authenticated information . The server returns the authentication result to terminal 1.
7	Terminal 1 keeps authenticated information. After the authentication ends, terminal 1 requests service
8	The server provides service.
9	The terminal 1 receives service.

Thus, the main feature of the proposed method is keeping authenticated information to be able to confirm the server has already authenticated terminal 1.

Next, Figure 8 shows and Table 7 explains the flow through which terminal 1 passes information that has already been authenticated to terminal 2 and through which terminal 2 receives service from the server.

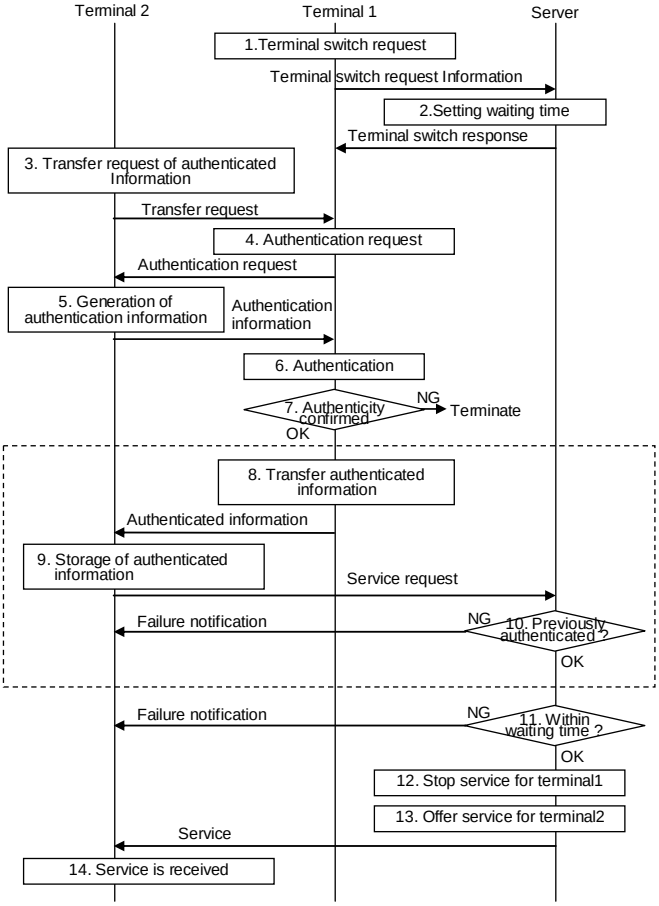

Fig. 8 General cooperation method for information that has been authenticated

TABLE VII
Explanation of Figure 8

No.	Explanation
1	Terminal 1 transmits the terminal switch request to the server.
2	The server sets waiting time.
3	Terminal 2 sends the forwarding request for previously authenticated information to terminal 1.
4	To confirm terminal 2 is not an illegal terminal, terminal 1 sends the authentication request to terminal 2.
5	Terminal 2 generates the authentication information in accordance with the request from terminal 1 and sends it to terminal 1.
6	Terminal 1 authenticates terminal 2.
7	Processing is interrupted if terminal 1 fails to authenticate terminal 2.
8	Terminal 1 transmits authenticated information to terminal 2.
9	Terminal 2 keeps the received authenticated information. Terminal 2 requests services using this information.
10	Terminal 2 (sender of authenticated information) has previously confirmed authenticity of the server.
11	The server confirms whether authenticity confirmed within waiting time set at step 2.

12	The server stops service to terminal 1.
13	The server provides service for terminal 2.
14	The terminal 2 receives service.

Thus, when terminal 2 receives a service by transmitting authenticated information from terminal 1 to terminal 2, the authentication processing on the server side can be simplified in the proposed method.

Contactless ICs, Bluetooth, Wi-Fi, or combinations of them can be used to send authenticated information between terminal 1 and terminal 2.

B. Mapping to actual protocol

We explained the general rules of the proposed method in section 3.1. In this section, we map actual protocols. When an protocol is mapped in the dashed line box in Figure 8, the flow should be analyzed. Other flow can be thought of as common flow that do not depend on an actual protocol. The flow in the dashed line box in Figure 8 is described in the following sections.

1) Federation method of information that has been authenticated using cookies

When the proposed method is applied to authenticating a terminal by using cookies, only information that the cookies have authenticated is needed. An example flow is shown in Figure 9 and explained in Table 8.

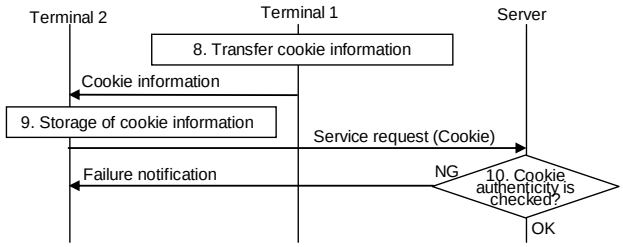

Fig. 9 Flow when proposal method is applied to authentication by using cookies

TABLE VIII
EXPLANATION OF FIGURE 9

No.	Explanation
8	Terminal 1 transmits cookie information to terminal 2.
9	Terminal 2 keeps received cookie information. and uses it to request a service .
10	The server confirms authenticity of cookie information that terminal 2 has transmitted.

2) Federation method of information that has been authenticated using GBA

When the proposed method is applied to authenticating a terminal by using GBA, only authenticated session key (Ks_NAF) information is needed. An example flow is shown in Figure 10 and explained in Table 9.

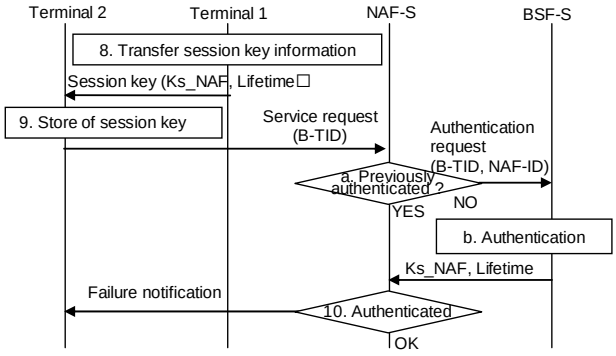

Fig. 10 Flow when proposal method is applied to authentication by using GBA

TABLE IX
EXPLANATION OF FIGURE 10

No.	Explanation
8	Terminal 1 transmits session key information (Ks_NAF, Lifetime) to terminal 2.
9	Terminal 2 keeps received session key information. Terminal 2 requests service with session ID (B-TID).
a	If terminal 2 cannot be confirmed to have been previously authenticated, the NAF server transmits B-TID and NAF-ID to the BSF server.
b	The BSF-S executes the authentication processing. The BSF-S generates Ks_NAF and sends Ks_NAF and lifetime to NAF-S.
10	The NAF-S confirms result of authentication that BSF-S has transmitted.

3) Federation method of information that has been authenticated using IPSec

When the proposed method is applied to authenticating a terminal by using IPSec, only authenticated SPD and SAD information is needed. An example flow is shown in Figure 10 and explained in Table 9.

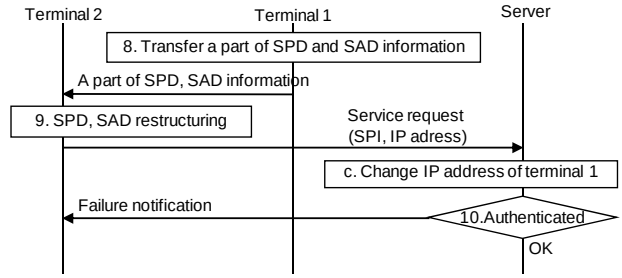

Fig 11 Flow when proposed method is applied to authentication by using IPSec

TABLE X
EXPLANATION OF FIGURE 11

No.	Explanation
8	Terminal 1 transmits a part of SPD and SAD information to terminal 2.
9	Terminal 2 restructures its DB on the basis of received SPD and SAD information. Terminal 2 transmits SPI and IP address to the server.
c	The server changes IP address of terminal 1 received at Step 9.
10	The server communicates by using SAD and SPD information that it is keeping.

IV. Implementation

In this chapter, we describe the prototype system that is implemented on the basis of the Cookie method proposed in section 3.2.1. After this chapter, we describe using a smart phone instead of terminal 1 and a PC instead of terminal 2.

A. Coordinated technology of smart phone and PC in Cookie method

In this section, we explain the coordinated function of Cookie information when the service transfers from a smart phone to a PC.

Figure 12 shows the proposed flow. Figure 12 is a more detailed version of Figure 9. (STEP1) First, a user taps a smart phone to PC. The PC module reads the ID of contactless IC chip in a smart phone by using contactless IC communication. (STEP2) Next, the PC module executes the pairing processing with the smart phone module. At this time, the conversion value of the above-mentioned contactless IC chip ID is used as PIN code for pairing Bluetooth. (STEP3) After completing the pairing, the PC side module requests Cookie Information by the Bluetooth communication. Afterwards, Cookie information is stored in the DB of the PC web browser. (STEP4) Afterwards, the PC web browser is started and accesses an Intra-net server using the above-mentioned Cookie information. The server transmits site information after checking Cookie information, and the PC browser displays the site.

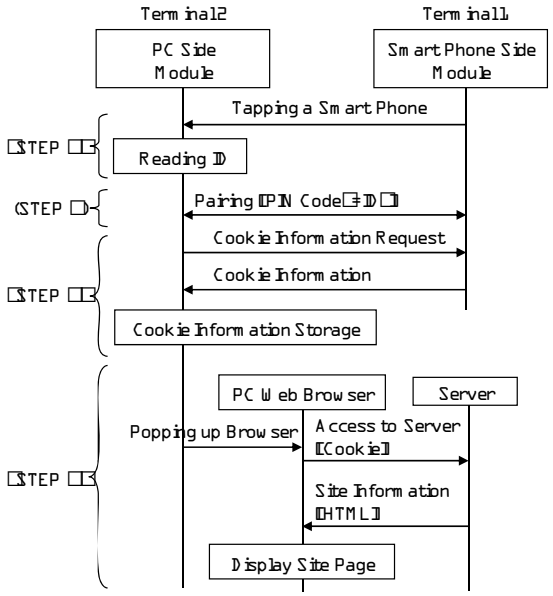

Fig. 12 Federation technology of smart phone and PC in Cookie method

B. Function of PC module

The PC module operates starting with reading the signal from the contactless IC reader. Afterwards, it does the Bluetooth pairing processing, receives Cookie information and URL from the smart phone module, and starts the browser processing. We next describe each function of the proposed system implemented this time.

(1) Contactless IC chip ID reading function

This function operates when a smart phone is held up to a contactless IC reader and acquires ID of the contactless IC chip.

(2) Bluetooth pairing function

This function pairs the two pieces of Bluetooth equipment (the PC and the smart phone) automatically. In the pairing, the 16-byte PIN code generated from ID acquired by reading the contactless IC chip is used[1].

(3) Cookie information import function

This function acquires Cookie information transmitted from the smart phone module and imports it to the PC browser. Cookie information is imported to Firefox by operating the Cookie database of Firefox directly in this implementation.

(4) The browser start function

This function displays URLs received from the smart phone module by starting Firefox in another process.

C. Smart phone module function

The smart phone module executes in the background. It enters the incoming connection waiting state from the PC module as soon as it starts. We next describe each function of the proposed system that was implemented this time.

(1) Contactless IC chip ID acquisition function

This function acquires ID of noncontact IC chip built into the terminal[2].

(2) Bluetooth pairing function

This function pairs the two pieces of Bluetooth equipment (the PC and the smart phone) automatically. In the pairing, the

[1] Preferably, the decision of the connection destination acquires the Bluetooth address with ID by ideally using the contactless IC chip ID reading function. However, this does not happen in this implementation.

[2] Preferably, this function acquires ID from the contactless IC chip built into the terminal. However, ID registered in the registry beforehand is acquired in this implementation.

16-byte PIN code generated from ID acquired by reading the contactless IC chip ID is used[3].

(3) Export function of Cookie information

This function acquires Cookie information that Internet Explorer Mobile manages, converts it into the data format for the Cookie transmission, and transmits it to the PC module.

(4) Acquisition and transmitting function of URL

URL displayed with Internet Explorer Mobile is acquired and transmitted to the PC side[4].

V. Evaluation of Execution Time

In this chapter, we evaluate the execution time by using the system developed in accordance with the proposal method.

A. Measurement condition

Tables 11 and 12 show specs of the terminal that measures the performance. Moreover, we used FeliCa for noncontact IC communication between terminals. We use the Bluetooth Ver.2.1 adaptor for the Bluetooth communication after the handover. In addition, we use the ADSL communication network of 8Mbps for the network between PC and the server.

TABLE XI
SPECS OF PC USED FOR MEASUREMENT

OS	Windows XP SP2
CPU	Intel Core 2 Duo T8100 (2.1GHz)
Memory	3GB
Browser	FireFox 3.0.7

TABLE XII
SPECS OF SMART PHONE USED FOR MEASUREMENT

OS	Windows Mobile 6 Standard Edition
CPU	ARM1136 OMAP2430(330MHz)
Memory	256MB(ROM)/128MB(RAM)
Browser	Internet Explorer Mobile

B. Measurement item

The processing of the measured object is as follows. Each processing phase corresponds to those in Figure 12.

- Time between tapping a smart phone to a PC and reading ID from the smart phone (STEP 1).
- Time between request to do pairing and pairing finishing (STEP 2).
- Time between sending Cookie request and storing Cookie on DB (STEP 3).
- Time between starting up a browser and completely showing the site (STEP 4).

C. Results of measurements

We show the results of measurements in Table 13. Each measurement item was measured 12 times, and the mean value was calculated from ten time measurements to which the entire time had excluded the minimum and the maximum data.

TABLE XIII
RESULT OF A MEASUREMENT

Measurement item	Result of a measurement (Sec)
STEP 1	0.31
STEP 2	0.76
STEP 3	1.49
STEP 4	1.64
Total	4.19

As understood from Table 13, we can confirm that service was received about four seconds after the smart phone was tapped to the PC.

VI. Conclusion and Future Work

We have developed an authentication method that simplifies the authentication processing between terminal 2 and the server by coordinating information that has already been authenticated between terminals 1 and 2. Specifically, we demonstrated the method can coordinate information collected by using three different authentication methods (cookies, generic bootstrapping architecture (GBA) method, and IPSec). In addition, we measured the execution time of the system that was implemented on the basis of the proposed method. We also showed that we can receive service about four seconds after we tap the smart phone to the PC.

We will develop the verification technology to verify the terminal omitted in this implementation. Moreover, we will advance implementation of the system on the basis of proposed methods other than the Cookie method in the future.

Acknowledgment

This study is a part of the result of "Research and development of the terminal platform technology" sponsored

[3] The terminal on the connected side should also specify the other party's address in the pairing of Bluetooth. In this implementation, a smart phone waits for the socket connection from the PC. After the connection, the Bluetooth address on the PC side is acquired from the socket connection information. Afterwards, the pairing processing is executed.

[4] Because the URL from Internet Explorer Mobile was not able to be acquired by this implementation, URL is hard coded to the registry.

by National Institute of Information and Communications Technology (NICT).

About Trademarks

References

[1] IETF RFC2109 HTTP State Management Mechanism
[2] IETF RFC2965 HTTP State Management Mechanism
[3] "Generic Authentication Architecture (GAA) Generic bootstrapping architecture," 3GPP TS 33.220 3rd Generation Partnership Project
[4] IETF RFC 2617 HTTP Authentication: Basic and Digest Access Authentication
[5] IETF RFC 2818 HTTP Over TLS
[6] IETF RFC 3310 Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA)
[7] IETF RFC 4301 Security Architecture for the Internet Protocol
[8] IETF RFC 4302 IP Authentication Header
[9] IETF RFC 4303 IP Encapsulating Security Payload (ESP)
[10] IETF RFC 4306 Internet Key Exchange (IKEv2) Protocol
[11] IETF RFC 4807 IPsec Security Policy Database Configuration MIB
[12] IETF RFC 4809 Requirements for an IPsec Certificate IETF Management Profile
[13] IETF RFC 4478 Repeated Authentication in Internet Key Exchange (IKEv2) Protocol
[14] IETF RFC 3526 More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)