Development Reports

FirstPass PC Support Functions

Hiroaki Yamamoto, Masaharu Nakatsuchi,
Masayuki Kanaya, Masashi Onogawa,
Yasuo Morinaga and Tetsuya Hiruta

"FirstPass," FOMA's digital authentication service, provides SSL client authentication in order to improve the security in i-mode communications. We developed a mechanism that allows using these functions for PC-based communication as well.

1. Introduction

As the usage of Internet-based applications such as Internet shopping, stock trading and remote access to corporate intranets using mobile terminals and PCs expands, maintaining a high level of security through personal authentication is becoming more and more important. The current situation, however, is that customers are using IDs and passwords as authentication means, in spite of the fact that they are afraid of, and complain about lack of security.

In the digital authentication service "FirstPass" [1] for FOMA (Freedom Of Mobile multimedia Access), which we started providing in July 2003, we adopt an authentication method where FOMA users can send client certificates to Contents Providers (CP) that are compatible with FirstPass. This allows Internet access with improved security and simpler operations than conventional authentication methods where multiple IDs and passwords must be managed for each service used by customers.

By confirming the client certificates validity received, the CPs can reduce risks such as "spoofing" by third parties, compared to conventional password authentication.

On the other hand, environments where client certificates can be used for user authentication have already been provided as a mechanism when using PCs for various applications. Examples of such mechanisms include Smart Card logon authentication used when logging on to Windows, authentication by Transport Layer Security (TLS) in the case of PPP Extensible Authentication Protocol (EAP) on a Wireless Local

Area Network (WLAN) used when connecting to a network, authentication by Internet Protocol security (IPsec) on a Virtual Private Network (VPN) and Secure Sockets Layer (SSL) [2] client authentication*1, which makes use of Internet browser technology when using applications. Moreover, directory servers and authentication servers such as Remote Authentication Dial-In User Service (RADIUS)*2 servers, which manage user information for customers, also implement mechanisms that allow them to work in connection with client certificates.

As discussed so far, the platforms for using client certificates are becoming mature, and there is a growing demand for digital authentication functions that provide high security and are easy to use in environments involving both mobile terminals and PCs.

For this reason, we have developed functions that allow FOMA terminals to use FirstPass' client certificates from external devices as well.

This article discusses issues and countermeasures when using FirstPass to link with PCs, the new functions equipped into FOMA, the corresponding PC software developed, as well as the services provided by these functions and software.

2. FirstPass Usage Technologies and Requirements for PC Applications

FirstPass uses the Public Key Infrastructure (PKI)*3, the most widely used encryption infrastructure on the Internet, and the client certificates used are defined in the ITU-T X.509 [3] standardized by the International Telecommunication Union-Telecommunication standardization sector (ITU-T). The client certificates themselves can thus be used on many platforms.

A client certificate, when the user him/herself uses it, is handled in a pair with a private key unique to the user. Since this private key can be used in various applications, including digital signature and encryption purposes, due attention is required for the storage and handling of the key. For this reason, devices that are easy to carry with high secrecy and tamper-resistant*4 are needed.

Currently, smart cards and Universal Serial Bus (USB) type memory keys are the most commonly used technologies for such applications, but they require resolving issues concerning the cost and convenience involved in card issuance and connection of dedicated readers.

FOMA terminals contain highly secure FOMA cards (hereinafter referred to as User Identity Modules (UIMs)) as attachments, and FirstPass client certificates and private keys are also stored in the UIMs. Moreover, FOMA terminals are equipped with USB interfaces as standard; they can be connected to PCs immediately, simply by inserting cables.

This means that it is possible to use a UIM as a smart card and a FOMA terminal as a reader hypothetically, thus solving the problems outlined above.

In order to allow using FirstPass from PCs as well, we require the following:

1) It must be possible to access the service from a PC browser in the same way as for i-mode.
2) Taking expandability and versatility into consideration, Application Programming Interfaces (APIs) that can be used in various applications must be made available and it must be possible to use standard applications without any modification.
3) The security policy of the system must support both convenience and a high level of security when handling the Personal Identity Number 2 (PIN2) code, which is the security code associated with the client certificate.

3. FOMA Terminals Functions for FirstPass PC Applications

3.1 Background for the Development

Considering the user convenience when adopting client certificate in PC applications as discussed in Chapter 2, we examined the necessary function implemented to FOMA terminals.

3.2 Newly Developed Functions for PC Applications

The SSL client authentication sequence used in web browsing applications is as follows. **Figure 1** shows a timing sequence diagram for the client authentication and command input processing. First, a FOMA terminal sends a "ClientHello" message to the server and handshaking is started. The server then requests a client certificate by sending a "CertificateRequest" message, to which the FOMA terminal first replies with a

*1 SSL client authentication: This is a communication method where a third party authentication body certifies that a given server (domain) on the Internet actually exists and, at the same time, authenticates the user as well, allowing both the user and server to exchange digital certificates issued at that time to perform communication. The data communicated between the user and server in this communication method is also encrypted.

*2 Remote Authentication Dial-In User Service (RADIUS): This is a protocol for performing authentication of network users and recording usage in a unified manner. The main functions are to judge whether or not to allow connection according to user information registered in a database and keeping track of connections; the protocol is defined by IETF RFC 2138.

*3 Public Key Infrastructure (PKI): This is an infrastructure technology using public key cryptosystem technologies and the standard security technology on the Internet. It is widely used, especially for authentication and digital signature applications using digital certificates. It is defined by the Public Key Infrastructure working group (PKIX) of the Internet Engineering Task Force (IETF).

*4 Tamper-resistant: This concept refers to a construction where confidential information stored inside a device cannot be extracted or tampered with by people other than the legitimate right holders.

"ClientCertificate" message containing the client certificate and then with a "CertificateVerify" message containing signed information. Upon receiving these messages, the server verifies the validity of the signature included in the sequence above using the public key included in the client certificate.

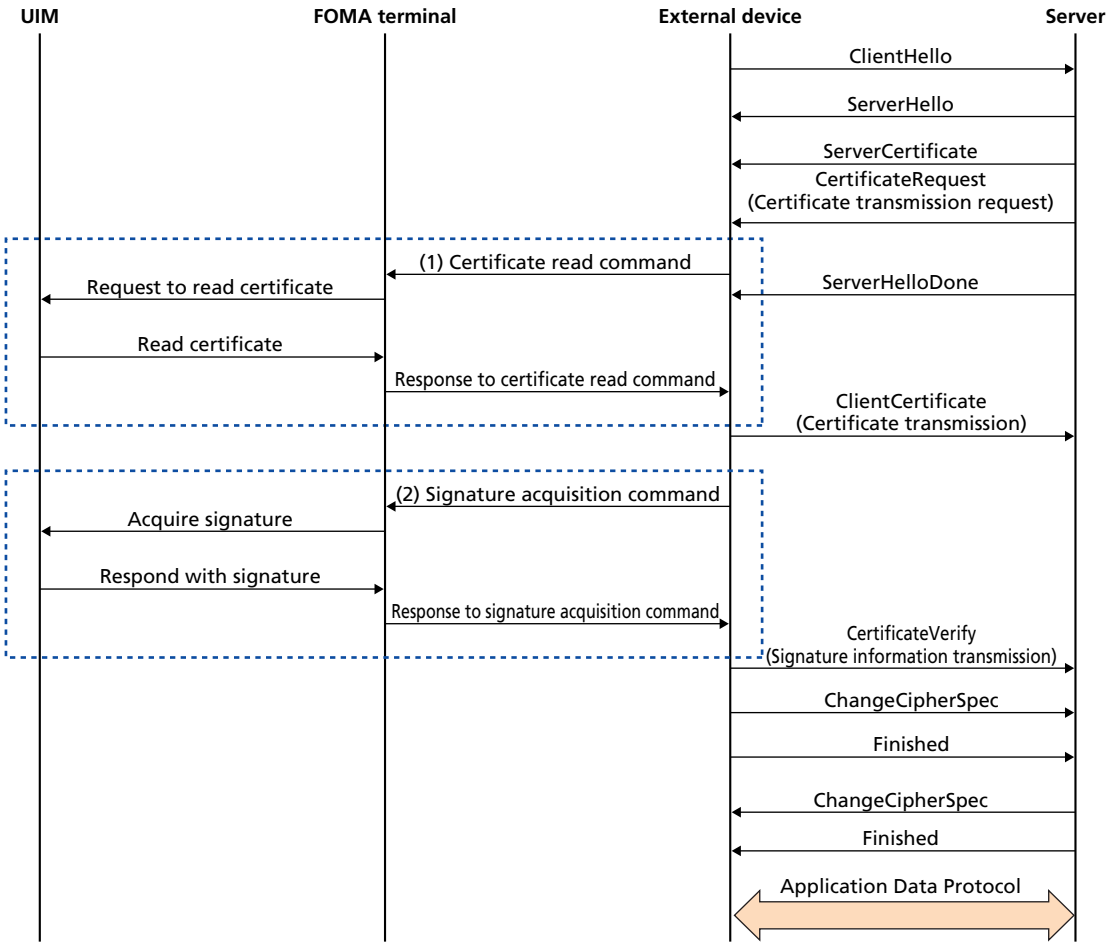
Following functions are equipped in order to perform the above-mentioned handshaking with an external device using the certificate and private key stored in the UIM within the FOMA terminal:

- Reading the certificate stored in the UIM and passing it to the external device
- Acquiring signed data using the private key stored in the UIM

In order to achieve the functions above, it is necessary to set up a new interface in the FOMA terminal. Furthermore, it is noted that certificates can be used in other contexts than just FOMA data communication; for instance, if the external device to which the FOMA terminal is connected is a PC with built-in WLAN hardware, the certificate can also be used in authentication when accessing the WLAN (EAP-TLS authentication). We thus created specifications to implement these two functions, which work no matter what the physical form of an external device might be, considering the fact that client certificates should be usable for FOMA and WLAN network independently, without depending on the bearer.

There exists a wide range of external interfaces that can be used as means of connection between an external device and a FOMA terminal, including connection via a USB cable, connection using Personal Computer Memory Card International Association (PCMCIA), a PC Card slot which works just like a data communication card, and connection via Bluetooth[*5]. Moreover, it is linking with external devices using various new

*5 Bluetooth is a registered trademark of Bluetooth SIG, Inc., USA.



(1) The external device sends an AT command requesting to read a certificate.
(2) Hash data and PIN2 code required to acquire the signature are sent.

: Newly developed function

Figure 1 SSL client authentication when using FirstPass certificate in PC environments (Example)

means of connection is assumed in the future. In order to cause minimum impact on the FOMA terminal development and to be able to use not depending on the physical form, we implemented function realization by ATtention (AT) command.

The AT commands are used for exchanging call control signals between the external device and the FOMA terminal as well as for setting parameters required for communication, which are widely used in PC communication. By using this well-established method, it becomes easier to develop control applications that allow using client certificates with PCs.

The behaviors of the two implemented commands are explained below (Fig. 1).

1) Certificate Read Command

We defined a command for reading the route certificate, sub-route certificate and client certificate that are stored in the UIM, so that the appropriate certificate can be sent in response to a certificate request from the server. Each certificate can be specified and read by passing a parameter along with the command.

2) Signature Acquisition Command

In order to sign a message using the private key stored within the UIM at authentication, it is necessary to send the data to which the signature should be applied to the UIM. This data is hashed data from the transmission of "ClientHello" at the start of client authentication to before transmission of "CertificateVerify." According to the UIM policy, it is also required to enter the PIN2 code in order to obtain the signature. Given these requirements, we enabled the function to send the hash data and PIN2 code at the same time as the command parameters when entering the command, sign the hash data using the private key stored within the UIM and transmit the result to the external device.

3.3 Problems Involved in Using Certificates with PCs

Since SSL communication takes place during web browsing, a FOMA terminal is required to continue communication to perform authentication during data communication.

The FOMA data communication uses the modem port for dial-up communication, and it is thus not possible to input commands during communication even if the modem port supports the AT commands. In order to solve this problem, we allowed usage of a command port that is not used in data communication, thus making it possible to "input SSL commands transparently to users" even during data communication.

However, to perform SSL communication smoothly, it is not enough simply to read the data. To use the data read via SSL communication, some dedicated middleware is required in order to allow monitoring the client authentication handshake processing, automatically inputting commands when the server requests presentation of client certificate and signature, and reading information within the UIM of the FOMA terminal and passing it to the server when requested. In order to achieve this, we made a prototype of a verification tool and confirmed that authentication can be performed by an external device transparently to the user. Thus, we confirmed that there were no problems in implementing the AT commands in FOMA terminals and concluded that it was possible to utilize client certificates with PCs.

4. Development of PC Software

In order to use a client certificate to perform SSL client authentication, it is necessary to operate the PKI function implemented in a FOMA terminal by means of AT commands.

This chapter provides an overview of the development of a Cryptographic Service Provider (CSP) module[*6] to perform SSL client authentication from a PC browser using the PKI function implemented in a FOMA terminal.

4.1 Software Functions Overview

Figure 2 shows the software functions overview.

Two functions must be provided in order to perform SSL client authentication from a PC.

One obtains a certificate from the FOMA terminal and registers it in the OS' certificate storage (certificate registration function), which is provided by a separate application independently of the CSP module. This application is activated manually by the user from the Start menu or similar and allows not only registration of a certificate but also other operations on the certificate such as deleting and adding, modifying and deleting friendly names, as well as setting communication with the connected FOMA terminal.

The other function, which is provided by the CSP module, handles signing data in response to signature requests from the web server (signature function). The CSP module is activated automatically in response to a signature request from the web

*6 Cryptographic Service Provider (CSP) module: Software consisting of modules providing cryptographic functions, digital signature generation functions etc. New modules can be created in addition to those defined by the standard. When creating such new modules, however, it is necessary to obtain an appropriate code signature from Microsoft in order to run them on Windows®.

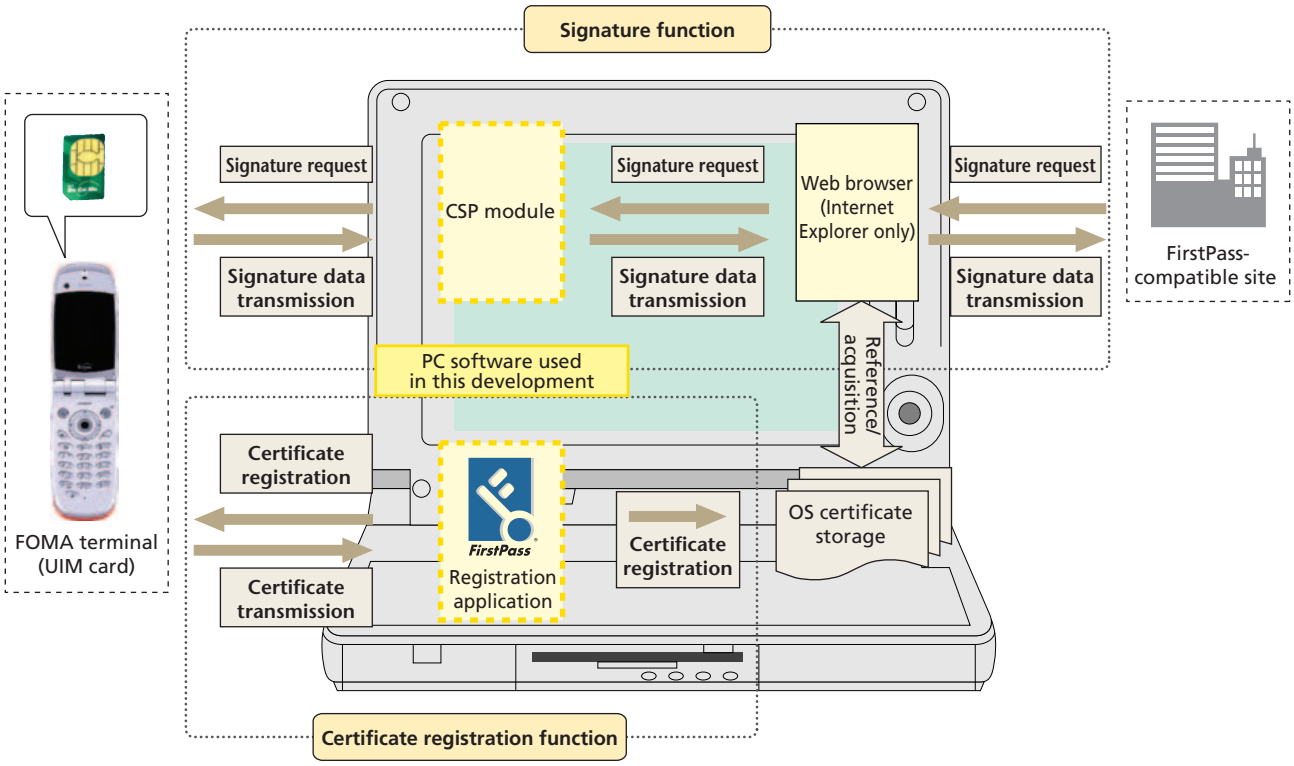


Figure 2 Overview of software functions

browser (Microsoft® Internet Explorer®*7 only); the user does not need to start it up manually.

4.2 Browser Support

In fact, all signature requests from the browser to the CSP module are performed from the API that provides cryptographic functions used by the browser. For this reason, the CSP module must have an interface with the API providing the cryptographic function; in other words, it must support the API providing cryptographic functions.

In Internet Explorer, the CryptoAPI*8 advocated by Microsoft is used to provide cryptographic functions, whereas PKCS#11 (Cryptoki)*9 of the Public Key Cryptography Standards (PKCS) is used for Netscape's Netscape® Navigator®. This means that for the CSP module developed this time to support these two browsers, it must support two different types of API providing cryptographic functions. This is synonymous with developing two types of CSP modules, each of which supports one of the APIs providing cryptographic functions. Since this would cause the development scale to become significantly larger, we only developed the CSP module that supports the CryptoAPI, i.e., Internet Explorer.

Moreover, we added versatility to the CSP module so that it can be used together with any application using the CryptoAPI in other authentication contexts than SSL client authentication, such as WLAN authentication (EAP-TLS) and VPN authentication (IPsec) as well.

4.3 Certificate Registration Function

When performing SSL client authentication from Internet Explorer, the certificates registered in the OS' certificate storage are used, instead of obtaining the certificate from the UIM every time the client certificate presentation is requested. For this reason, it is necessary to obtain the certificate from the UIM and register it in the OS' certificate storage in advance. The certificate registration function was developed for this purpose.

This function is explained below, along with the user operations involved (**Figure 3**).

1) The user activates the application from the Start menu or similar and clicks the [Register] button in the certificate management tab to send the certificate read command to the

* 7 Microsoft® and Microsoft Internet Explorer® are registered trademarks or trademarks of Microsoft Corporation in USA and other countries.

*8 CryptoAPI: An API developed by Microsoft that provides cryptographic functions to an application. It consists of three layers: an application layer, a system layer and a CSP layer.

*9 PKCS#11 (Cryptoki): PKCS refers to a collection of APIs providing cryptographic functions developed by RSA Laboratories in collaboration with representatives from the industry, academia and the government. Among them, PKCS#11 (also known as cryptoki) is designed so that a device referred to generally as cryptography token, can be logically displayed with an application.

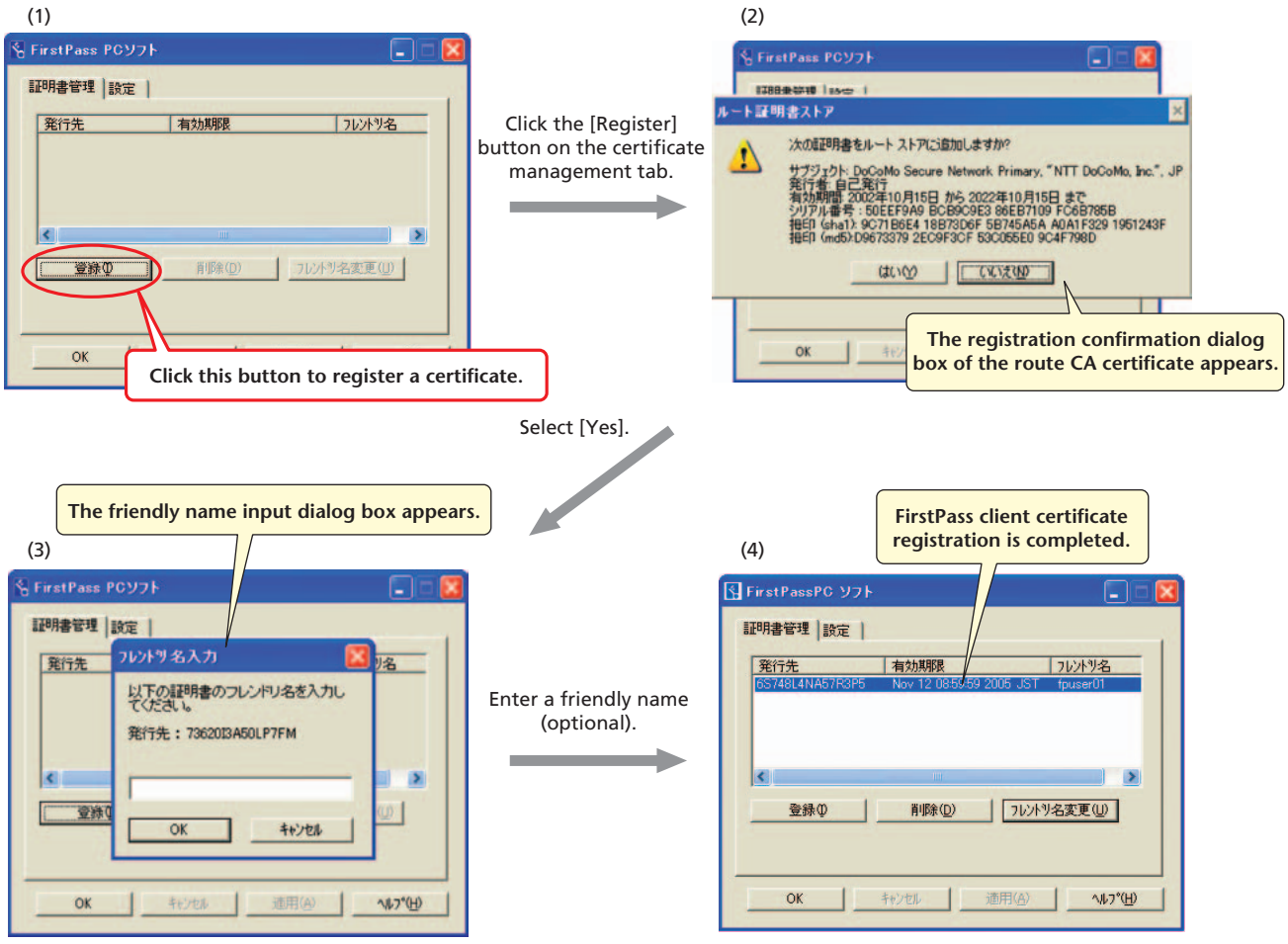
UIM and acquire the necessary certificates (client certificate, intermediate Certification Authority (CA) certificate and route CA certificate).

2) The user registers the three certificates acquired in step (1) in the OS' certificate storage. At this point, if the route CA certificate to be registered does not exist in the certificate storage, a dialog box confirming whether or not to register a new CA certificate appears (this dialog box does not appear if the route CA certificate to be registered already exists in the certificate storage).
3) Next, the friendly name input dialog box appears. A friendly name is a nickname given by the user to a certificate, and can simply be omitted. Friendly names can be checked not only in the certificate management dialog box but also in the certificate dialog box of Internet Explorer.
4) The user enters a friendly name and clicks the [OK] button (click [OK] without entering anything if the friendly name is to be omitted) to complete the certificate registration and make the certificate ready to be presented when Internet Explorer requests the client certificate.

When registering certificates, the private key stored in the UIM together with the client certificate is not acquired and sent to the PC, thereby preventing the private key from falling into the hands of third parties.

4.4 Signature Function

When the presentation of the client certificate is completed, the CSP module is invoked by the CryptoAPI in order to append a signature. The CSP module then sends the data to be signed together with the signature acquisition command to the UIM, and the data is signed using the private key stored in the UIM. Moreover, since inputting the PIN2 code is mandatory to obtain access to the UIM for security purposes, the PIN2 code is also sent when the signature acquisition command is sent to the UIM. The CSP module is invoked by the CryptoAPI and per-



* Screen shots reprinted with permission from Microsoft Corporation.
** This function is provided in Japanese only.

Figure 3 Flow of certificate registration

forms these operations automatically in the background (if entering the PIN2 code is required, an input dialog box appears to prompt the user to enter it).

The flow of operations in the two functions discussed in Sections 4.3 and 4.4, as seen from the user side, is shown below (**Figure 4**).

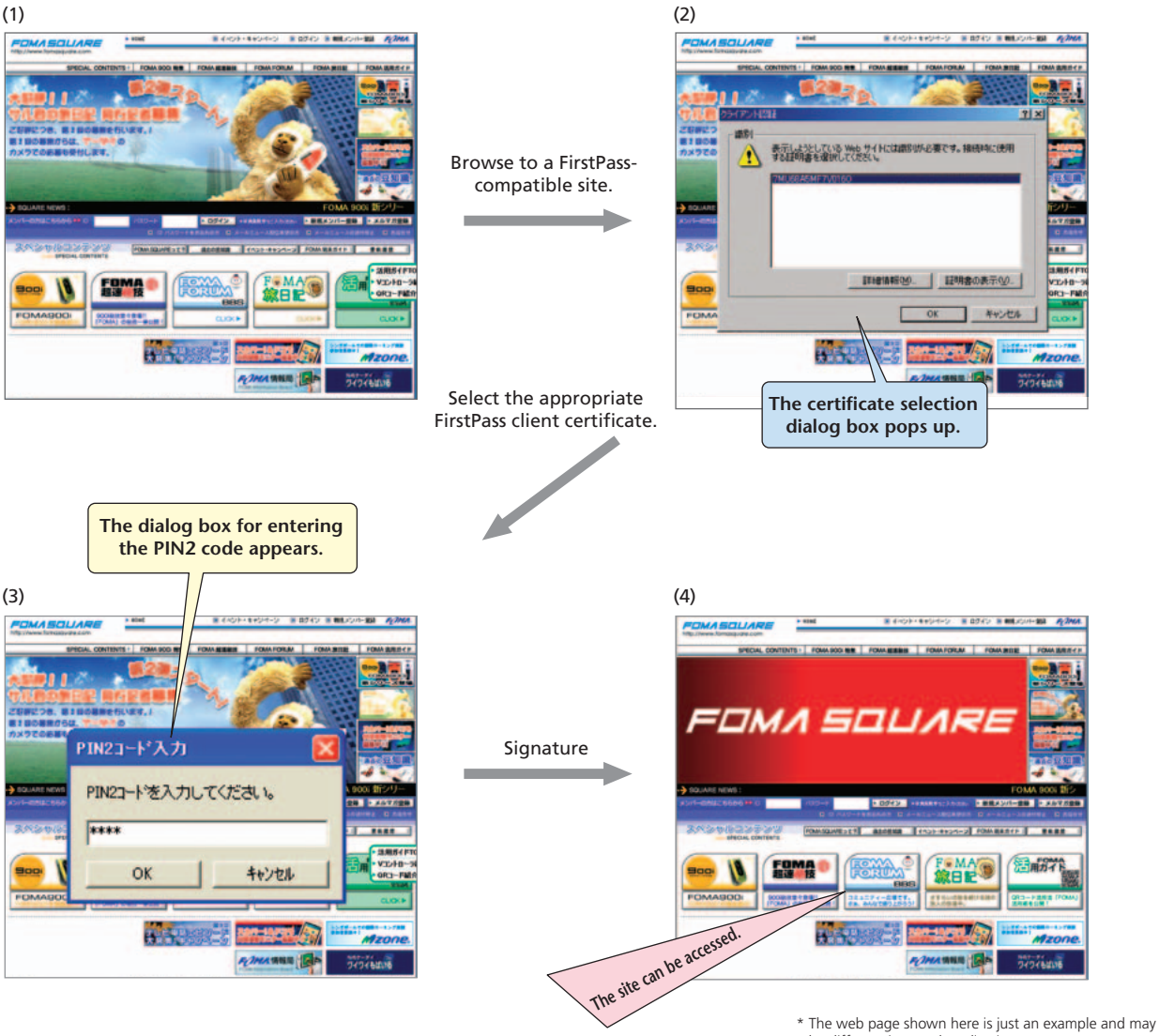
1) The user accesses a FirstPass-compatible site using Internet Explorer.
2) When the site (web server) is accessed, it requests the client certificate. Upon receiving the request, Internet Explorer displays the certificate selection dialog box and prompts the user to select a certificate.
3) When the client certificate is sent, an input dialog box appears, in which the user enters the PIN2 code necessary to access the UIM in order to append the private key signature in the UIM.
4) When signing is completed and authentication is completed, the site can be accessed.

4.5 Functions for Improving Usability

In order to improve the usability, two additional functions are implemented in the developed software. One of them is the "active FOMA terminal specification" function and the other is the "PIN2 code keeping" function.

First, we explain the "active FOMA terminal specification" function.

Normally, both the modem port and the command port can be used when communicating with a FOMA terminal, but there are situations where one of them may not be available; for example, the modem port cannot be used during dial-up com-



* The web page shown here is just an example and may be different in actual applications.
** This function is provided in Japanese only.

Figure 4 Flow of SSL client authentication processing

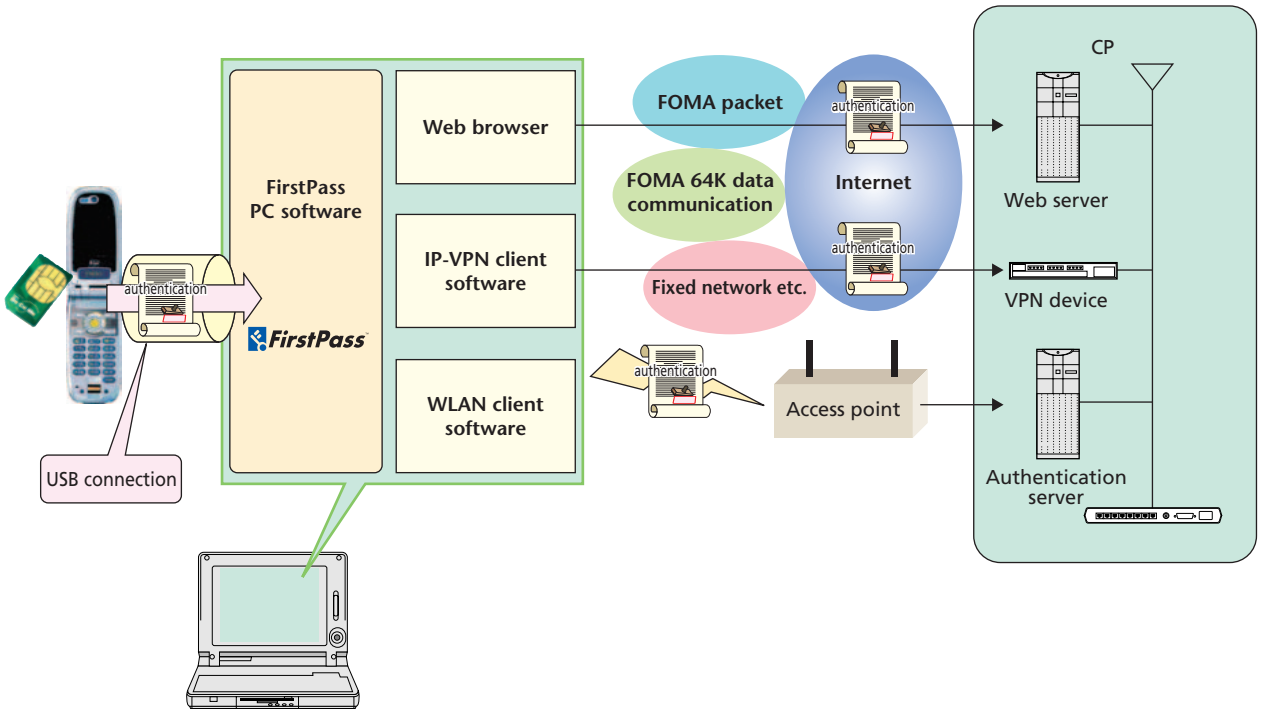


Figure 5 Utilization of FirstPass in PC applications

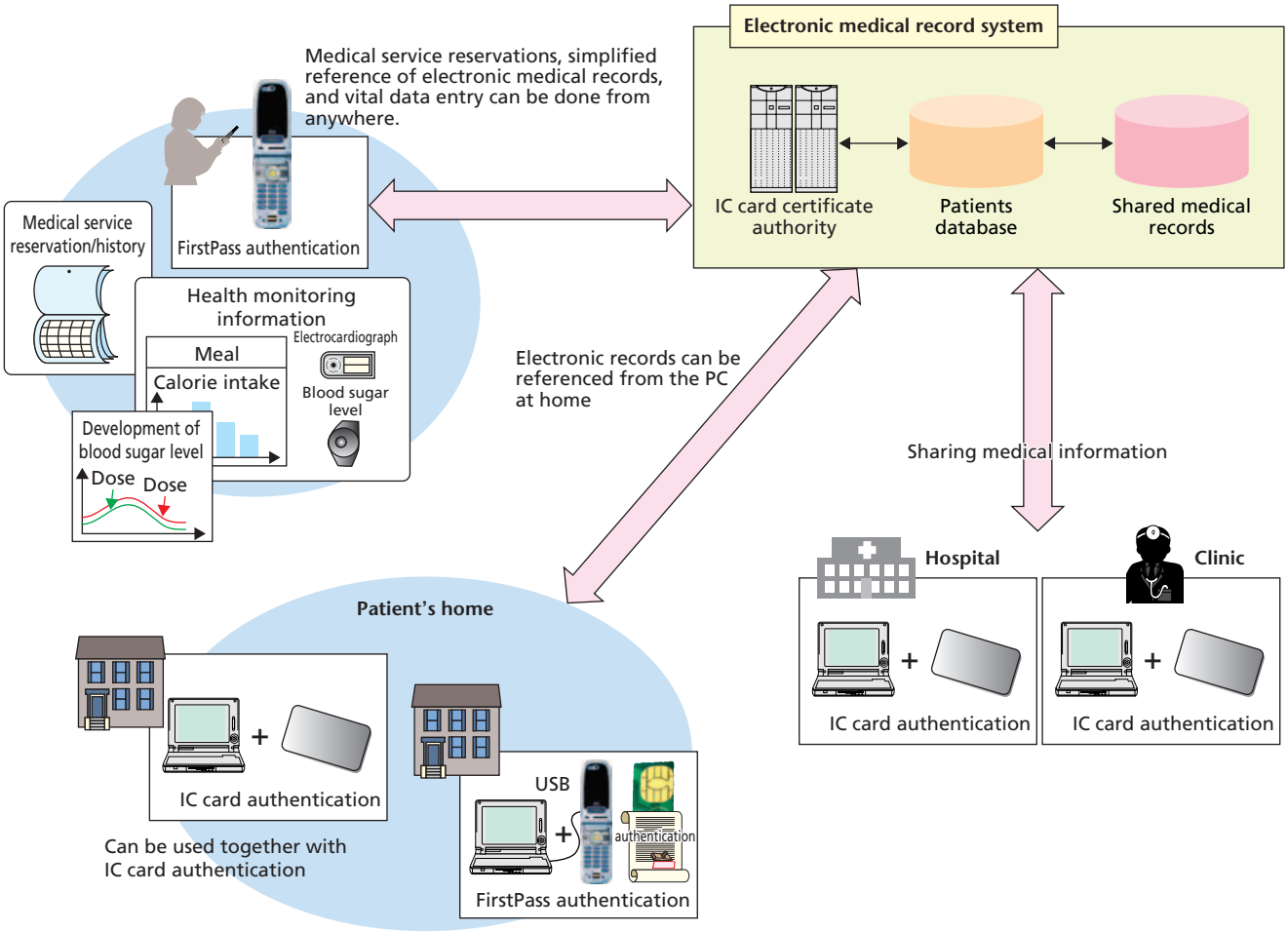


Figure 6 Usage example (medical service)

munication. For this reason, it is necessary to check which port can be used in the actual application of the software, which complicates the operations. The "active FOMA terminal specification" function solves this problem.

This function automatically selects the port that can be used, either the modem port or command port, and acquires the corresponding COM port number when the FOMA terminal to be used is selected. If both ports can be used, the command port is chosen by default.

Next, we explain the "PIN2 code keeping" function.

This function temporarily memorizes the PIN2 code in order to reduce the trouble for the user of inputting the PIN2 code many times. Recording the PIN2 code in a file and similar, however, may represent a significant security risk, so the code is kept in memory only. We furthermore designed the function so that all PIN2 codes kept in memory are deleted in case of Windows logoff and shutdown operations. This function also allows users who place high importance on security to delete codes from memory automatically or invalidate the keeping function itself when a predefined time elapses.

The PIN2 code is encrypted when kept in memory; it will thus never be possible for third parties to decode the PIN2 code.

5. Conclusion

As discussed above, the development of the FOMA terminal functions and PC software allows using the FirstPass client certificate for a wide range of authentication purposes, including SSL client authentication, WLAN authentication (EAP-TLS authentication) and VPN authentication (IPsec), without having to rely on any communication network (**Figure 5**). This makes it possible to perform local authentication, such as Windows logon, as well as network authentication as far as the client certificate downloaded from a FOMA terminal is available. For the customers, the convenience is improved because devices such as IC cards and authentication tokens are replaced with a FOMA terminal, and in addition, the cost is reduced as there is no need to purchase new devices in order to improve the security. In the future, it is expected that this technology will be applied to various services requiring a high level of security, such as access to corporate intranets, payments services and medical services (**Figure 6**).

References

[1] N. Nakamura et al.: "Special Articles on FirstPass Digital Authentication Service," NTT DoCoMo Technical Journal, Vol.5 No.3, PP.4–23, Dec. 2003.
[2] A. O. freier. P. Karlton and P. C. Kocher: "The SSL Protocol Version3.0"
[3] A. Arsenault and S. Turner: "Internet X. 509 Public Key Infrastructure Roadmap," IETF draft-ietf-pkix-roadmap-09, IETF PKIX Working Group, Jul. 2002.

Abbreviations

API: Application Programming Interface
AT: ATtention
CA: Certification Authority
CP: Contents Provider
CSP: Cryptographic Service Provider
EAP: PPP Extensible Authentication Protocol
FOMA: Freedom Of Mobile multimedia Access
IETF: Internet Engineering Task Force
IPsec: Internet Protocol security
ITU-T: International Telecommunication Union-Telecommunication standardization sector
PCMCIA: Personal Computer Memory Card International Association
PIN: Personal Identity Number
PKCS: Public Key Cryptography Standards
PKI: Public Key Infrastructure
PKIX: Public Key Infrastructure working group
RADIUS: Remote Authentication Dial-In User Service
SSL: Secure Sockets Layer
TLS: Transport Layer Security
UIM: User Identity Module
USB: Universal Serial Bus
VPN: Virtual Private Network
WLAN: Wireless Local Area Network