Special Articles on FirstPass Digital Authentication Service

Technology for the Implementation of PKI Functions in Mobile Terminals

Kazuhiko Takahashi, Hiroyuki Sakakibara,
Makoto Hamatsu, Nobuyuki Watanabe,
Chiaki Nogawa and Chie Noda

In providing the FirstPass service, an SSL client authentication function is implemented in the F2102V and N2102V FOMA mobile terminals, and a simpler and more secure terminal authentication function by servers is realized than that of conventional terminal.

1. Introduction

Secure Sockets Layer (SSL) [1] is a secure communication protocol that is widely used on the internet. At DoCoMo we have already incorporated an end-to-end server authentication function in the 503i series of digital mobile terminals Personal Digital Cellular (PDC) and in Freedom Of Mobile multimedia Access (FOMA) terminals. This function makes it possible to authenticate servers and carry out cryptographic communications. However, conventional server authentication was under a constraint that although terminals can authenticate servers, servers were not able to authenticate terminals. Therefore, for the F2102V and N2102V FOMA terminals and the User Identity Module version 2 (UIMv2) (FOMA card (green)) (**Photo 1**), we developed functions for SSL client authentication (hereinafter referred to as client authentication), and started the service called FirstPass. FirstPass is Japan's first commercial service for mobile communications in which client authentication is implemented based on the Public Key Infrastructure (PKI). In FirstPass client authentication, the terminal presents the server with a user certificate and a signature during SSL handshaking, allowing the IDs allocated to each FOMA contract

Photo 1 FirstPass compatible terminals (F2102V, N2102V) and the UIMv2 FOMA card

to be authenticated. Since the server and client can thereby authenticate each other, it is expected that FirstPass will be applicable to a wider range of services.

To perform client authentication, it is necessary to have a function for obtaining a user certificate in advance, and a function for presenting this certificate together with a signature during client authentication.

In this article, we describe the implementation of PKI functions in terminals and User Identity Modules (UIMs) that are needed to achieve client authentication.

2. The Requirements and Issues of PKI Functions in Terminals

2.1 PKI Function Requirements

What is required for client authentication is a user certificate which is obtained from the DoCoMo Certification Authority (CA). When a server requests client authentication during SSL communication, the terminal presents the user certificate and a signature to the server. Accordingly, the terminal must support the following functions:

- User certificate obtaining functions (key-pair generation, issue application, downloading).
- Functions for presenting the user certificate and generating/presenting a signature during client authentication.

The user certificate obtaining functions must generate a pair of keys when applying for the certificate issuance. Since the private key is used to generate a signature, this key must be generated and stored at the terminal in a secure manner. Also when operating the user terminal, the application to DoCoMo CA for user certificate and downloading it must be performed securely.

To generate a signature for client authentication, the terminal must have a function that can generate a signature without disclosing the private key.

2.2 The Issues of Conventional Terminals

To satisfy the PKI function requirements, it is important to consider how the certificate and key pairs are managed. In conventional terminals, since there were no PKI functions in the User Identity Module version 1 (UIMv1) (FOMA card (blue)), it was thought that the terminal memory would be used to generate and store the key pairs and to store the certificate. However, this not only causes security risks, but also makes it impossible to continue using the same private key and certificate when the terminal is updated. As a result, client authentication functions were not implemented in these terminals.

3. PKI Functions in FirstPass-compatible Terminals

To resolve these issues and satisfy the two requirements in section 2.1 for FirstPass-compatible terminals and UIMv2 cards, we implemented SSL client authentication by performing the PKI functions (e.g., the management of key pairs and certificates) in the UIM card to use the relevant functions from the terminal. Since there were no standard schemes that can be implemented in the PKI functions of the UIM card and interface between the terminal and the UIM, we developed our own solution.

As **Figure 1** shows, a terminal is a system that consists of a UIM, a certificate download application, and an i-mode browser. A UIMv2 card supports key-pair generation, certificate storage and signing functions (section 3.1). The certificate download application has functions to connect to the DoCoMo CA, apply for the issue of a user certificate in conjunction with the UIMv2, and download this certificate to the UIMv2 card (section 3.2). The i-mode browser includes an SSL protocol stack to

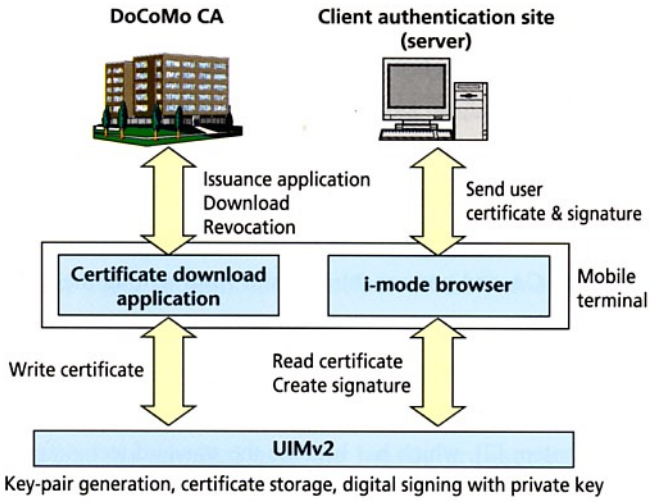

Figure 1 System configuration

protect the HyperText Transfer Protocol (HTTP) data, and has a function for sending the user certificate and a signature in conjunction with the UIM during client authentication (section 3.3). The implementation of each PKI function is described below.

3.1 The PKI Functions of UIMv2

To perform SSL client authentication, the PKI functions implemented in UIMv2 must mainly include the following three elements:

- Key-pair generation and private key storage
- Storage of a user certificate
- A digital signing function (public-key cryptography with a private key)

1) Key-pair Generation and Private Key Storage

It takes several tens of seconds to generate a key pair inside an IC card, and if this is done dynamically during communication with the DoCoMo CA, then it could lead to problems such as communication time-outs. For this reason the UIMv2 does not create key pairs, but instead it is pre-loaded with key pairs when shipped from the UIM manufacturer. Therefore, when sending a certificate issue application to the DoCoMo CA, a key pair is generated by a dummy process involving the retrieval of pre-loaded key pair. Due to constraints on the memory capacity available inside the UIM card, the number of key pairs that can be pre-stored is limited to five.

An IC card is a tamper-resistant[*1] data storage medium, and it is impossible to read the private keys from outside of the card. Furthermore, when performing computations that use the private keys, the user is required to input a Personal Identity Number (PIN) 2 code, thereby ensuring that user authentication cannot be achieved when the terminal is used illegally by a third party.

2) User Certificate Storage

The UIMv2 card has a space for storing the user certificate, and this certificate is associated with a private key corresponding to the public key contained within it. Security is maintained by allowing the user certificate to be updated only by accessing the DoCoMo CA, to prevent updates by parties other than DoCoMo CA and prevent third parties from writing their own illegal certificates.

3) Signing Function

For secure communication, FirstPass uses RSA[*2] public-key cryptosystem [2], which has become the standard technique on the internet. The key length of the key pairs stored in the UIM card is 1024 bits, and the RSA public-key cryptography computation is performed inside the UIMv2. **Table 1** shows the main parameters of the client authentication function.

Table 1 Main parameters of the SSL client authentication function

Supporting protocol	SSL V3
Public key cryptography algorithm	RSA (UIM)
Key length	1024bits (UIM)
Hash algorithm	SHA1, MD5

3.2 Certificate Download Application

The certificate download application connects to the DoCoMo CA when applying for the issue of a user certificate, downloading it, and revoking it when no longer required. Since this application supports HTTP browser functions, and data exchange with the DoCoMo CA is based on HyperText Markup Language (HTML) content, the content can be created and updated easily. To allow a certificate to be downloaded, the HTML format is partially extended to include protocol conversion of the commands and responses transmitted between the DoCoMo CA and the UIM. By establishing an SSL session with the DoCoMo CA, data is exchanged between them via HTTP with SSL protection. This application is accessed by clicking on "User Certificate Operations" in the i-mode menu, whereupon it connects with the DoCoMo CA and displays the FirstPass menu list.

The user obtains a user certificate by performing an issue application to initiate a download.

1) Issue Application Function

Figure 2 shows the user certificate issue application sequence. When the user starts up the certificate download application by clicking on "User Certificate Operations", this application connects to the DoCoMo CA and displays the FirstPass menu list. When the user clicks on "Request your certificate", a certificate issue request is sent after performing server authentication, and a key-pair generation command and a command for the creation of an issue application request are transmitted from the DoCoMo CA (④). The terminal transfers the key-pair generation command to the UIM, and a pair of keys (public key/private key) is generated inside the UIM (⑤). The terminal then produces a signed issue application request in Public Key Cryptography Standards (PKCS) #10 [3] format. First, it creates an issue application request including the public

*1 Tamper-resistant: Confidential information stored inside a device cannot be disclosed nor tampered with by anyone other than legitimate right holders.

*2 RSA: Public key cryptosystem. "RSA" are the initials of the three scientists who invented the system, R. Rivest, A. shamir and L. Adleman.

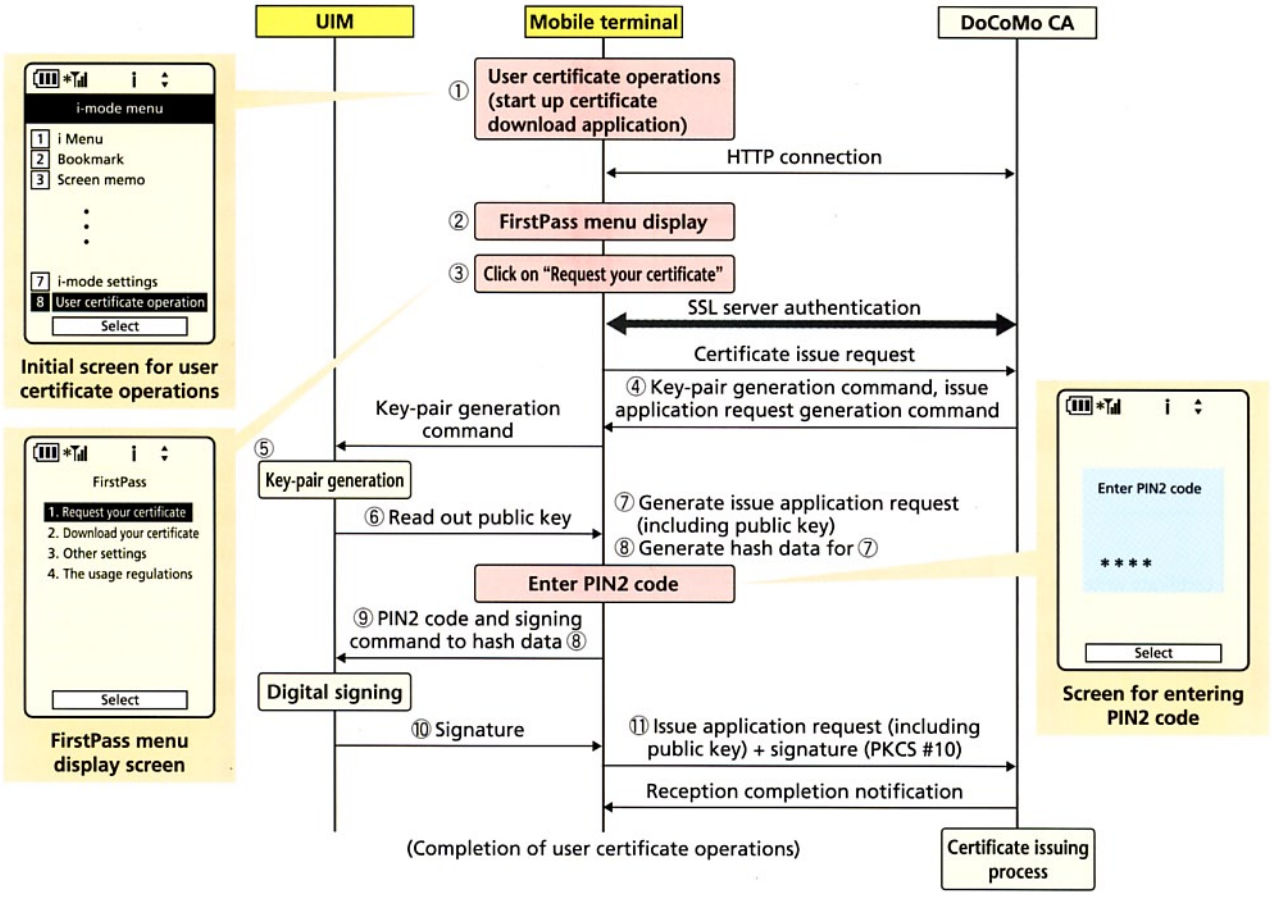

Figure 2 Certificate issue application sequence

key (⑥) read out from the UIM (⑦). Then, the hash data (⑧) of the same request is created, and for the signature, the hash data is performed computations with the private key inside the UIM (⑩). Finally, these are both changed into PKCS #10 format and sent out (⑪). Note that in the private key computations, it is necessary to check the PIN2 code in the UIM, so a screen for the input of the PIN2 code is displayed and it is input by the user.

When a signed issue application request that has been created in this way is transmitted to the DoCoMo CA, a reception completion notification and a downloadable period are displayed, and the certificate download application terminates.

2) Download Function

Figure 3 shows the download sequence performed after an issue application has been completed. When the user clicks on "Download your certificate" in the FirstPass menu list, the details of the user certificate to be downloaded from the DoCoMo CA are displayed. After the user has confirmed the content and pressed the "Continue" button, the downloading starts. A certificate write command to the UIM sent from the DoCoMo CA is then transferred to the UIM by the terminal, whereby the user certificate is downloaded to the UIM.

3) Revocation Function

When the user clicks on "Other settings" followed by "Revoke your certificate" in the FirstPass menu list, client authentication is performed with the DoCoMo CA. After that, the DoCoMo CA presents the user certificate details currently in use, and when the user clicks on "Continue", the corresponding user certificate is revoked. By allowing the certificate revocation by operating the terminal in this way, the user certificate can be revoked quickly and easily according to the user's will.

The client authentication process performed during revocation is similar to the process performed in the i-mode browser as described below.

3.3 Client Authentication in the i-mode Browser

The client authentication functions of the i-mode browser are implemented by expanding on the existing server authentication functions. **Figure 4** shows the handshake sequence performed during client authentication. To initiate the SSL handshaking, the terminal uses the i-mode browser to send a "ClientHello" message to the server (client authentication site). Unlike the server authentication sequence, the presentation of

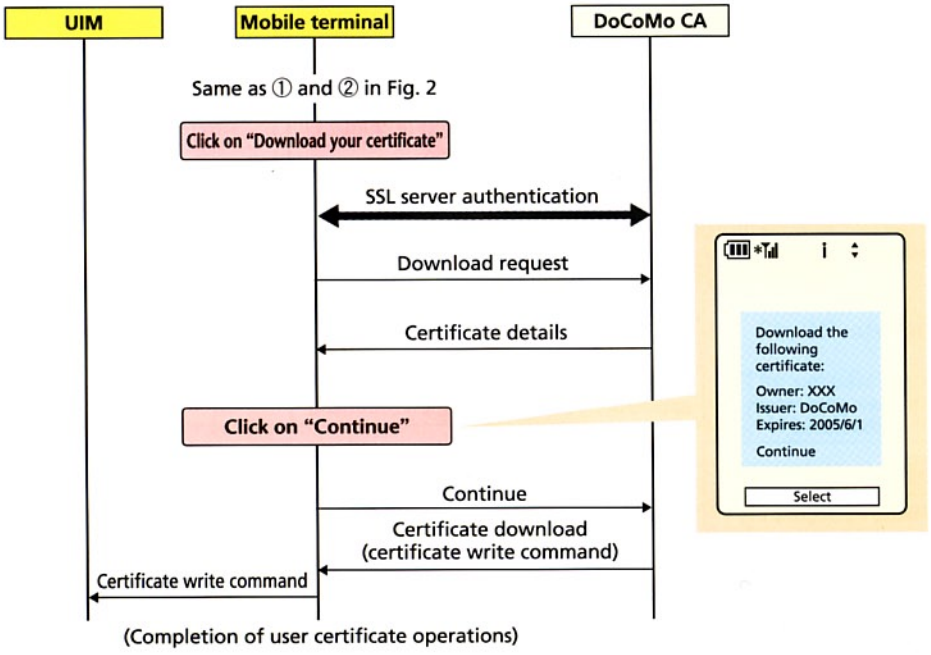

Figure 3 Certificate download sequence

the user certificate and signature is requested by the server. When the server transmits a "CertificateRequest" message ① requesting the user certificate, the terminal transmits the user certificate "ClientCertificate" ② and the signature information "CertificateVerify" ③. The server uses the public key in the user certificate to check the validity of the signature in the "CertificateVerify" message. In this way, the signature of the user certificate's owner is verified, making it possible to authenticate the IDs allocated to each FOMA contract in the user certificate (client authentication).

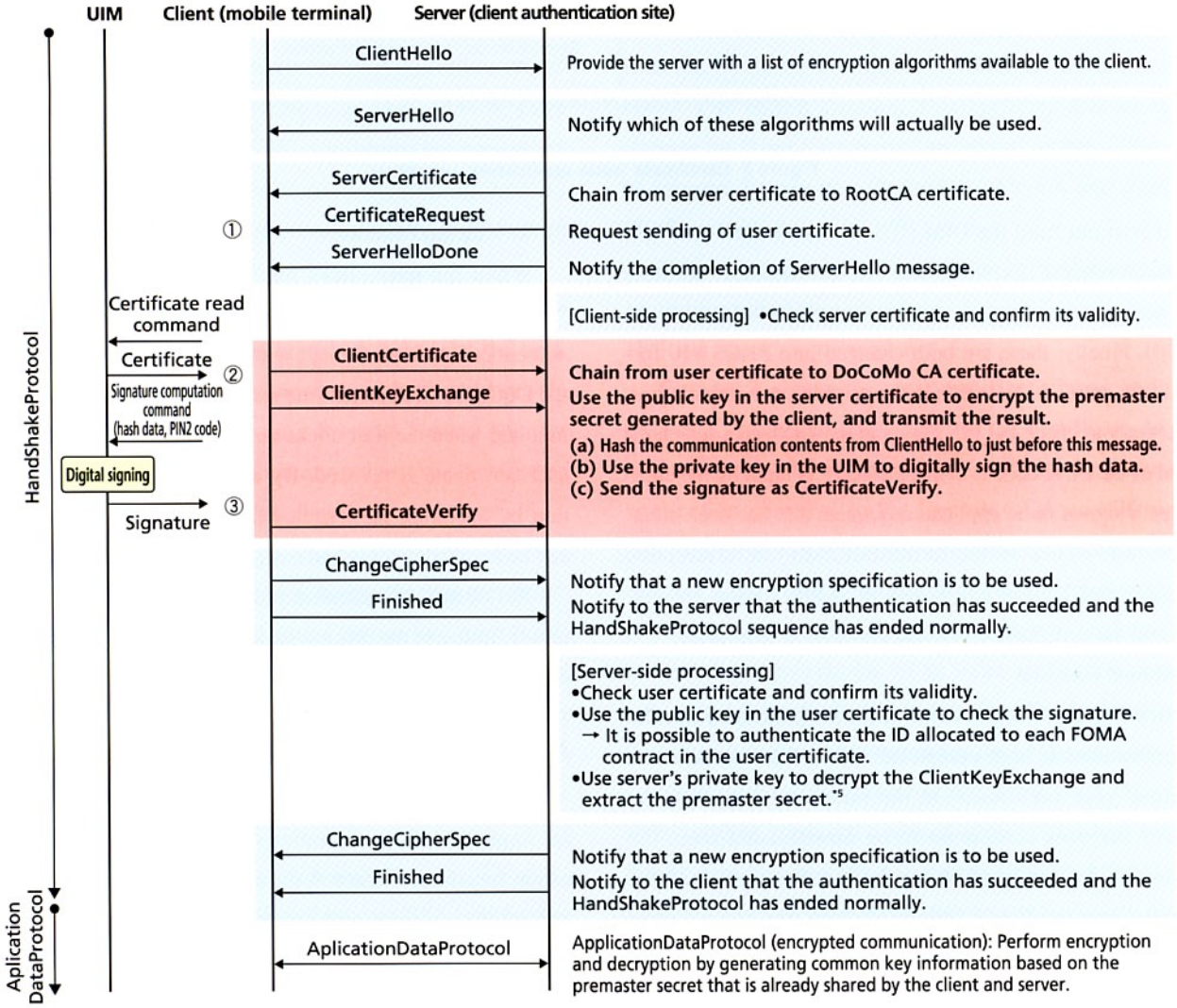

Figure 4 SSL client authentication protocol sequence

When transmitting the user certificate, "Your certificate is requested. Send your certificate?" message is displayed along with confirmation buttons (OK/Cancel), prompting the user for confirmation. If the user selects OK, the user certificate is sent.

When creating the "CertificateVerify" message, the terminal hashes*3 the protocol data and the like that was exchanged before the "CertificateVerify" message. It then transfers this hash data*4 to the UIM, which uses the private key to generate a digital signature for this data. The terminal transmits the signed data to the server as a "CertificateVerify" message. Note that the user is asked to enter the PIN2 code to perform the signing process in the UIM.

After handshaking, the HTTP data is encrypted, allowing secure HTTP communication with the server.

4. Conclusion

The FirstPass service includes tools for the secure downloading of a user certificate and SSL client authentication with a terminal and UIM card. This is a highly significant development in that it has made it possible to implement PKI-based client authentication solutions in mobile communications. Topics for further study include building up of more advanced authentication services by applying the user certificate and signature functions in the UIM card to other applications.

*3 Hashing: The compression of original data into fixed-length data by a one-way hash function such as SHA1 (Secure Hash Algorithm 1) or MD5 (Message Digest 5).
*4 Hash data: Data obtained as a result of hashing.
*5 Premaster secret: Data that forms the basis of each security parameter in SSL communication.

REFERENCES

[1] A.O. Freier, P. Karlton and P.C. Kocher: "The SSL Protocol Version 3.0", draft-freier-ssl-version 3-02.txt, Nov. 1996.
[2] RSA Laboratories, PKCS #10: Certification Request Syntax Version 1.5. Mar. 1998.
[3] B. Kalishki: RFC 2314: PKCS #10: Certification Request Syntax Version 1.5. Mar. 1998.

ABBREVIATIONS

CA: Certification Authority
FOMA: Freedom Of Mobile multimedia Access
HTML: HyperText Markup Language
HTTP: HyperText Transfer Protocol
MD5: Message Digest 5
PDC: Personal Digital Cellular
PIN: Personal Identity Number
PKCS: Public Key Cryptography Standards
PKI: Public Key Infrastructure
SHA1: Secure Hash Algorithm 1
SSL: Secure Sockets Layer
UIM: User Identity Module
UIMv1: User Identity Module version 1
UIMv2: User Identity Module version 2