● Development Reports ●

New Applications in FOMA 902i

Tatsuro Oi, Makoto Ueda,

Eiji Yano and Tomoyuki Tamechika

DoCoMo has developed four new applications for FOMA 902i terminals: ToruCa, SD-Binding, digital signatures, and individual charging.

1. Introduction

Along with significant advancement of application function on mobile terminal, many users are calling for greater convenience.

To help satisfy this need, we have expanded the functionality of several existing functions for use on the FOMA 902i terminals. We targeted, in particular, the FeliCa®*1 function, content protection for the Secure Digital (SD) memory card, and terminal security. As an extension to FeliCa, we developed a function that enables the capturing, display, and distribution of digital coupons and other digital items and data. For content protection on the SD memory card, we developed a mechanism that enables content to be used only under specific conditions thereby promoting the diversification of protected content. And, as an extension to terminal security, we developed a function that enables the generation of digital signatures on the mobile terminal itself.

We also developed a new function that enables content to be charged on an item-by-item basis in response to the trend toward a more diversified charging system.

This article presents an overview of these four functions newly developed by DoCoMo.

2. ToruCa Function

2.1 Service Concept

The FOMA 902i terminals are being equipped with the ToruCa function to promote the use of DoCoMo's "Osaifu-Keitai" (electronic wallet) service and the firm integration of this service into the life infrastructure. "ToruCa" refers to digitized forms of store coupons, membership cards, and other data

*1 Felica®: A contactless smart card technology developed by Sony and a registered trademark of the same.

that are presently distributed in paper form in the life infrastructure. (These digital forms are called "ToruCa data" below.) It is also a general term for services that distribute ToruCa data and display that data on mobile terminals. ToruCa data can be stored and displayed on mobile terminals via FeliCa, browsers, mailers, external interfaces, and such. The following outlines the ToruCa function for use on mobile terminals.

2.2 Capturing ToruCa Data

ToruCa data can be downloaded by holding one's mobile terminal in front of a Reader/Writer (R/W) unit equipped with the FeliCa function, or by using a browser as in DoCoMo's Chaku-melo and Deco-mail-template services. The data can be classified into two main types according to the bit rate of the transmission circuit used. These are "ToruCa snip," which provides just the minimally necessary amount of information for the item in question (a subset of available data), and "ToruCa in-detail," which provides all of the information for that item.

1) Capturing ToruCa Data by FeliCa

Holding one's mobile terminal in front of a FeliCa R/W initiates communications between the mobile terminal and FeliCa chip and automatically captures and stores ToruCa data. The mobile terminal notifies the user that data transfer has been completed by blinking a Light Emitting Diode (LED) or by ringing a sound. The communication system between the mobile terminal and FeliCa chip uses a FeliCa proprietary protocol specified by FeliCa Networks, Inc. In this protocol, ToruCa data is stored in an operator-specific data area, and the characteristics of this protocol are such that transmission data size is limited, which restricts transmission to the ToruCa-snip data type. And to further reduce the data size of a ToruCa snip, it is transmitted in binary data format. On the mobile terminal, however, ToruCa data is stored in only one data format regardless of the transmission path used for capture, and for this reason, a ToruCa snip in binary data format will be converted to text data format within the mobile terminal after capture.

2) Capturing ToruCa Data with a Browser

When downloading ToruCa data using a browser, the user will first be presented with a preview display of that data for confirmation purposes and will then be prompted to save the data. Capturing data by browser in this way differs from the FeliCa proprietary protocol described before in that there is no need to consider data-size limits, i.e., the target of capture can be either a ToruCa snip or ToruCa in-detail. Also, to make the maximum use of the terminal display performance, a text format based on Hyper Text Markup Language (HTML) can be used. **Figure 1** shows an overview of this format.

2.3 Displaying ToruCa Data

ToruCa data is a electronic coupons and other card-shaped items, it is therefore desirable that ToruCa data be displayed in a card-like manner. It is also necessary for the same ToruCa data be displayed in the same way on different terminal models, and to satisfy this requirement, the same ToruCa viewer has been installed on all models in the 902i terminals. **Figure 2** shows the viewer.

Since a ToruCa snip can only be displayed in plain text, the viewer is provided with a "show details" function that can cap-

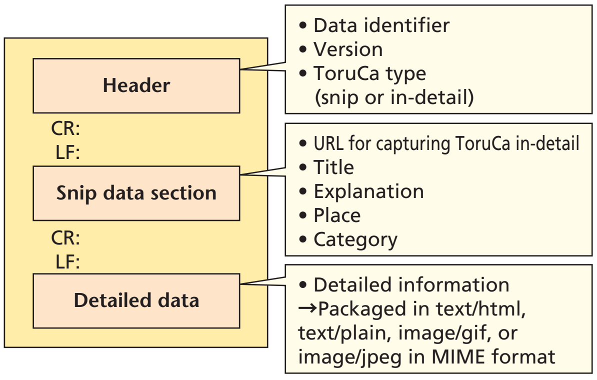
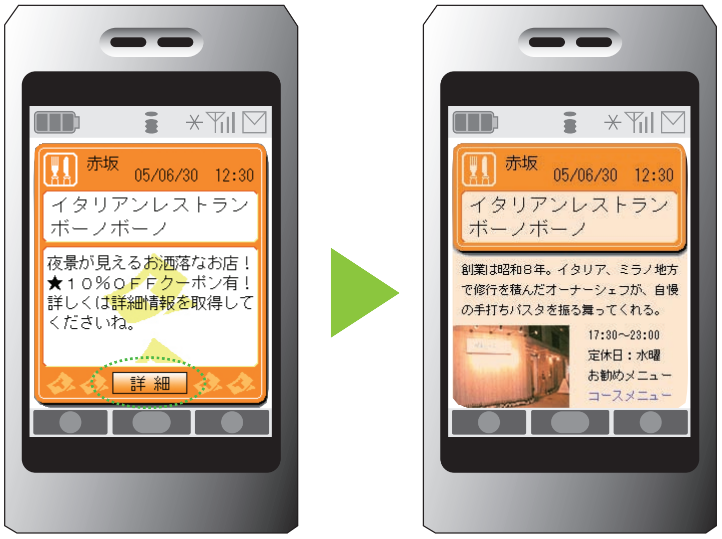


Figure 1 Overview of ToruCa data format



赤坂 05/06/30 12:30
イタリアンレストラン
ボーノボーノ
夜景が見えるお洒落なお店！
★１０％ＯＦＦクーポン有！
詳しくは詳細情報を取得してくださいね。

赤坂 05/06/30 12:30
イタリアンレストラン
ボーノボーノ
創業は昭和８年。イタリア、ミラノ地方で修行を積んだオーナーシェフが、自慢の手打ちパスタを振る舞ってくれる。
17:30～23:00
定休日：水曜
お勧めメニュー
コースメニュー

*This viewer is provided in Japanese only.

Figure 2 Displaying ToruCa viewer

ture ToruCa in-detail from the Uniform Resource Locator (URL) included in the ToruCa snip as an information element. This function can be easily executed by the user by pressing the "show details" button on the screen at the time of a ToruCa-snip display.

Images and text in HTML format can be used when displaying ToruCa in-detail for a richer expression compared to a ToruCa snip. Linking to other functions such as PhoneTo, WebTo, and MailTo can also be achieved through the use of HTML tags.

2.4 Exporting ToruCa Data

ToruCa data can be exchanged between mobile terminals by exporting as an e-mail attachment or exporting to an infrared connection or external memory. The export function on 902i terminals, however, allows the export of only ToruCa snips. An attempt to export ToruCa in-detail will result in the extract and the export of the ToruCa snip from that data. As a consequence, the mobile terminal on the import side will have to execute the "show details" function described above to recover ToruCa in-detail.

3. SD-Binding Function

3.1 Service Concept

For content or other files encrypted and stored in SD memory, the SD-Binding function embeds the conditions necessary for extracting the key used in encryption. This makes it possible to restrict the decoding of files to only those situations in which certain conditions (e.g., FOMA card, terminal model, i-appli) are met.

The SD-Binding function therefore enables the provision of various types of applications for reading and writing content from and to SD cards. For example, a content copyright-management function could be provided in accordance with the wishes of an existing i-mode Contents Provider (CP) to enable the downloading of that content regardless of memory restrictions on the mobile terminal, i.e., without having to delete useful data due to space limitations. The SD-Binding function also enables a smooth transfer of content when the user upgrades to a new mobile terminal (or is given a replacement mobile terminal due to operation problems). In addition, using statements like "i-appli's created by the same CP" in the method for specifying conditions can be used to create new services such as file sharing among several different i-appli.

Encrypting and storing i-appli data in SD memory also enables CPs to take full advantage of the large-capacity feature of SD cards. This capability will enable the provision of large-capacity i-appli's such as games that fully exploit map data and i-motion, sound effects, pictures, and saved data.

The next section explains bind types.

3.2 Bind Types

Five types of conditions have been defined in the SD-Binding function for use when encrypting content. We first take a look at two of these types that can be used when saving i-mode content. These are the User Identity Module (UIM) bind, which consists of user-specific information found in the FOMA card, and terminal-model bind, which is set with the model ID and the above UIM (**Figure 3**). A CP can specify these conditions for each content item at the time of download in the Hyper Text Transfer Protocol (HTTP) header that is attached to the response. The UIM bind allows content to move between different terminal models, while the terminal-model bind reduces the risk of losing content when expanding the terminal's storage capacity or when given a replacement mobile terminal due to problems with the original mobile terminal.

In addition to the above two conditions used with i-mode content, the following three conditions can be defined for accessing data used by i-applis. These are the "series bind" that uses terminal series data such as 902i for encrypting data, the "appli bind" that permits access only to the i-appli that saved the data, and the "CP bind" that only accepts the condition that the i-appli providers are identical. These conditions may be used in combination. The use of these bind types do not simply enable greater data capacities for i-appli data, but they also allow multiple i-appli's to access the same data. For example, a mail application could be allowed to access the data of a phone book application, and a large-scale application could be achieved by combining small-scale applications.

3.3 Other Functions

When encrypting content and storing it in SD memory, the specifications for the Content Protection for Recordable Media (CPRM) and SD-Binding standard specifications can be used to realize moving and sharing of contents that uses SD-card for media, between different manufacturers.

Also, as a scheme different from those described above, file type, title, and other content-related information can be affixed

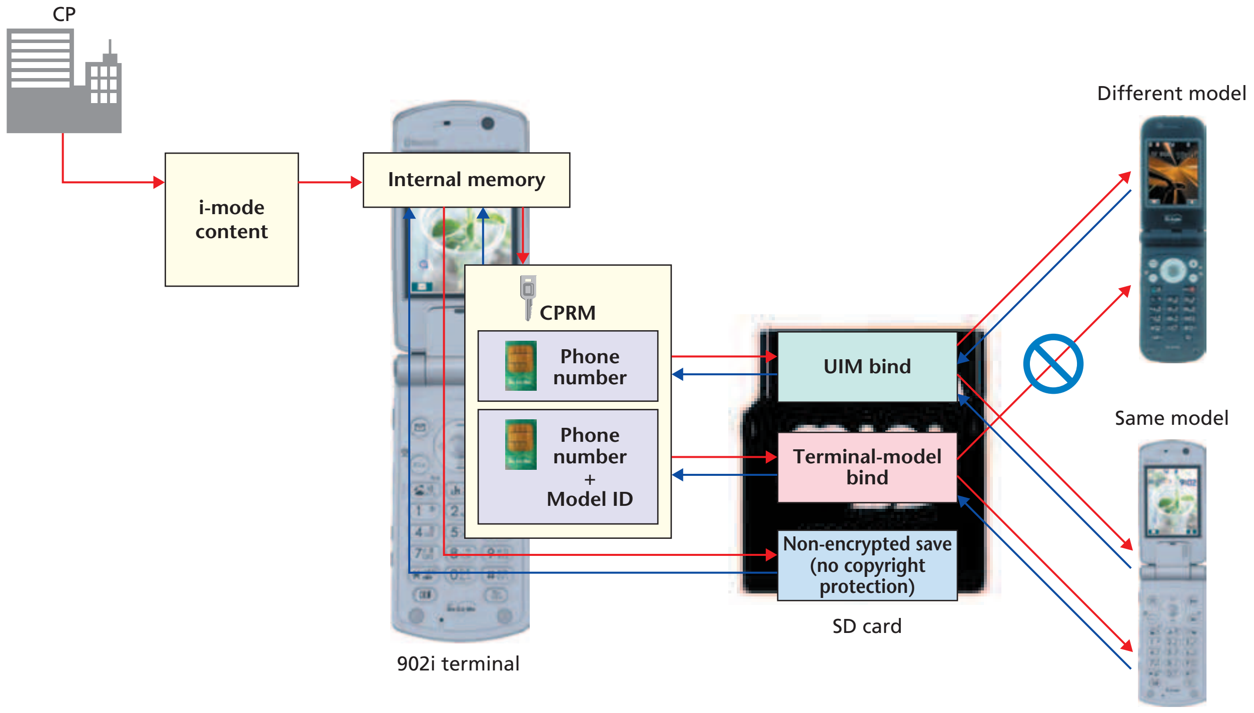


Figure 3 Bind types in SD-Binding

to content at the time of encryption in the form of a header. This would enable at least some information to be conveyed to the user, even for contents that are not available for some reason. It would simplify the identification of unnecessary files and make searches on SD cards more efficient.

In addition to simple encryption, a variety of attribute information possessed by each item of content such as ringtone-setting enable/disable can be stored in a protected area on an SD card inaccessible from a PC and other devices. Such a scheme would enable information not a part of content itself to be identified and processed as needed. It would enable, for example, playback-restriction management to be performed after encrypting content with such playback restrictions.

4. Digital Signature Verification and Generation by Java Applications

4.1 Service Concept

This service provides a digital-signature verification and generation function for transaction data (HTML, pictures, etc.) exchanged through communications between a mobile terminal and CP. These functions make it possible to detect tampering in transaction data as an addition to the provision of encrypted communications via Secure Sockets Layer (SSL) protocol and person authentication by an existing digital authentication service.

4.2 Function Conditions

The digital-signature generation function is achieved here by using a private key provided by FirstPass, which is a digital authentication service developed for FOMA terminals. Also, as communication services desired by CPs or other corporations can be provided by installing Java™[*2] applications, this function as well is provided by Java (i-appli) to give users a greater degree of freedom in its use.

4.3 Digital Signature Functions

1) Verification Function

Given that a CP has included a digital signature in transaction data sent to a mobile terminal, this function enables the user to authenticate the CP and to check for the presence of tampering in the transaction data all on the mobile terminal itself.

The above authentication process makes use of a Root Certificate Authority (RootCA) certificate that is preinstalled in the mobile terminal at the time of purchase as a trust anchor that is a basic point to establish reliability. The subject information in this RootCA certificate is compared with the subject information and issuer information of both the CP certificate and the intermediate certificate included in the digital signature sent

*2 Java™: Java and all Java-related trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

with the transaction data. This makes it possible to check whether each of the above certificates has been appropriately issued by a higher certificate and to authenticate the identity and existence of the CP certificate (**Figure 4** (1)-(3)).

Tampering in transaction data can be detected in the following procedure. Using the public key of the CP certificate included in the digital signature, a comparison is made between the value obtained by decrypting the digital signature value and the value obtained on the terminal side from the transaction data using the same hash function[*3]. If the values are the same, it can be deemed that no tampering has been done.

The digital-signature format that can be verified by this function conforms to the widely used PKCS#7SignedData format, and the digital-signature algorithms that can be verified are sha-1withRSA, md5withRSA, and md2withRSA.

2) Generation Function

A digital signature can be generated on the terminal side in response to transaction data received from a CP or other corporations, and that digital signature can be affixed to that transaction data steps (Fig. 4 (4)).

The private key that is paired with the FirstPass client certificate is used here to generate the digital signature. This key is stored in the UIM card, which is especially tamper-resistant. In addition, the process used to read out the private key is so designed that it can only be performed from a DoCoMo supplied application and not from general i-applis to prevent the leaking of secret informations. Moreover, to prevent threats associated with the prediction of a private key from large volumes of calculation results obtained by illegally generating digital signatures a huge number of times based on that key, a mechanism whereby the user of the mobile terminal in question must input a PIN2 code is adopted.

The digital signature format generated here conforms to the widely used PKCS#7SignedData format to make it easy for CPs and other corporations to perform digital-signature verification.

*3 Hash function: An operation that generates a pseudorandom number of fixed length from an input message. A hash function is generally a one-way function, which means that the original message cannot be obtained from the output hash value and that it is extremely difficult to create a different message having the same hash value.

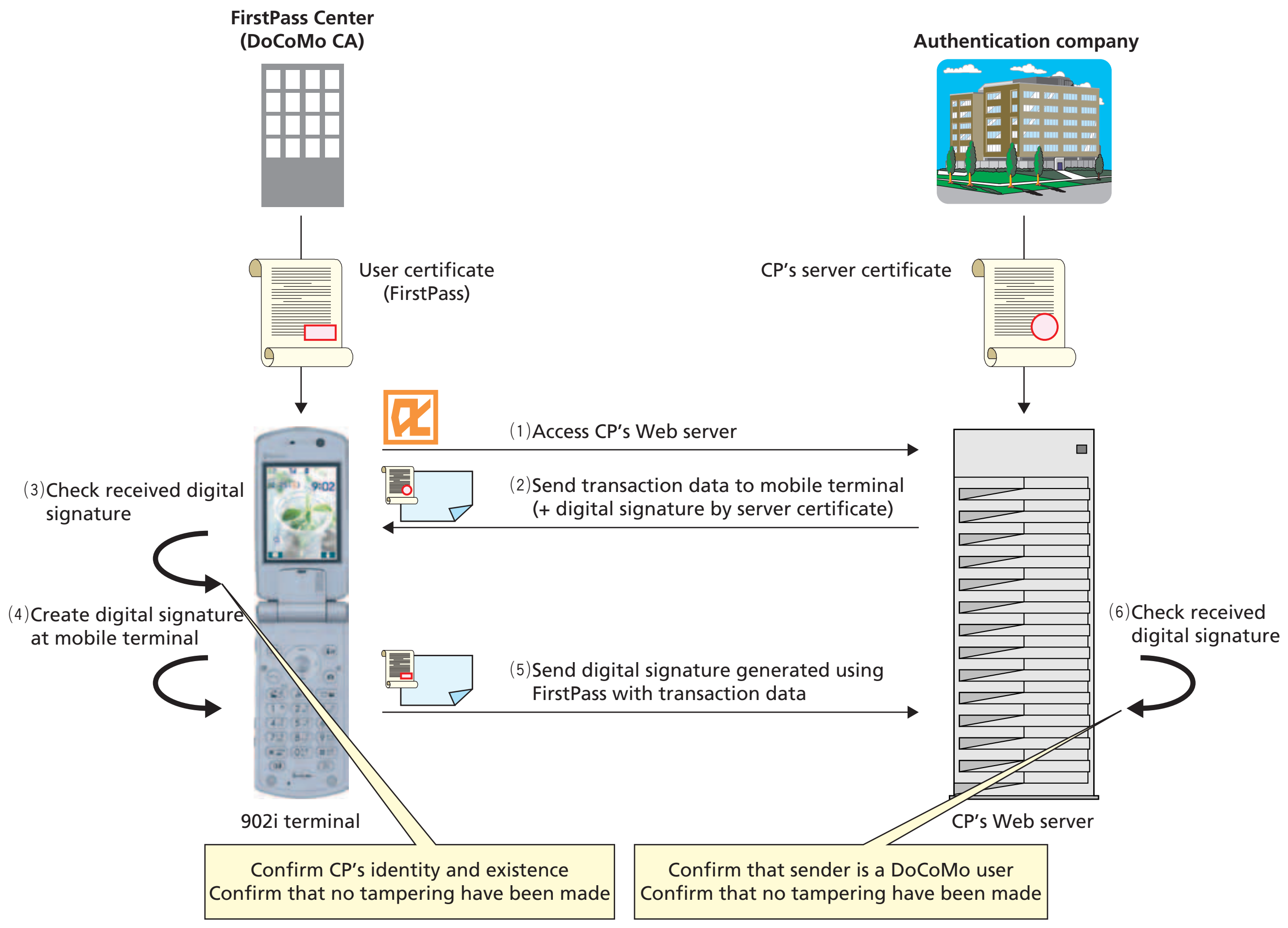

FirstPass Center
(DoCoMo CA)
Authentication company
User certificate
(FirstPass)
CP's server certificate
(1)Access CP's Web server
(2)Send transaction data to mobile terminal
(+ digital signature by server certificate)
(3)Check received digital signature
(4)Create digital signature at mobile terminal
(5)Send digital signature generated using FirstPass with transaction data
(6)Check received digital signature
902i terminal
CP's Web server
Confirm CP's identity and existence
Confirm that no tampering have been made
Confirm that sender is a DoCoMo user
Confirm that no tampering have been made

Figure 4 Digital-signature verification and generation

The digital-signature algorithms used in this generation process are sha-1withRSA and md5withRSA.

The mobile terminal attaches the generated digital signature to the received transaction data and returns the data to the Web-server side, which verifies the digital signature (Fig. 4 (5), (6)).

5. Individual Charging Function

5.1 Service Concept

On performing a comprehensive evaluation of current market and user trends in the wake of introducing a flat-rate packet communications system, DoCoMo is introducing a mechanism to enable content-by-content charging in addition to site-by-site monthly charging, the current charging system for content in existing i-mode services. The aim here is to promote increased use of i-mode services.

5.2 Individual-charging Mechanism

Figure 5 shows the sequence and screen transitions that make charging for individual items of content possible.

1) Using the terminal browser, the user accesses the target site, initiates the process for downloading the target content (password verification, etc.), and finally captures the download screen (Fig. 5 (1)).
2) The tag for performing "Download" in Fig. 5 (1) contains the attribute for switching over to "individual-charging mode." The terminal will discriminate the attribute and recognize that it starts downloading in individual charging mode, when user presses the "Download" button or anchor.
3) On downloading in individual-charging mode, an extension header will be attached to the request signal from the mobile terminal based on instructions received from the server side (Fig. 5 (2)). The server checks this extension header to confirm that downloading is being performed normally in individual-charging mode. In this regard, existing terminal models may sometimes display content while downloading is in progress, i.e., may display content as it is being received. The individual-charging function, however, sends a charge-notification response to the mobile terminal after downloading is completed, which means that partial content displayed in the above manner would enable content browsing before completion of the charging process. To avoid this situation, captured content is not allowed to be displayed until downloading and charging are completed (Fig. 5 (3)). Content acquisition in which no individual-charging extension header is attached corresponds to ordinary content acquisition with no individual charging.

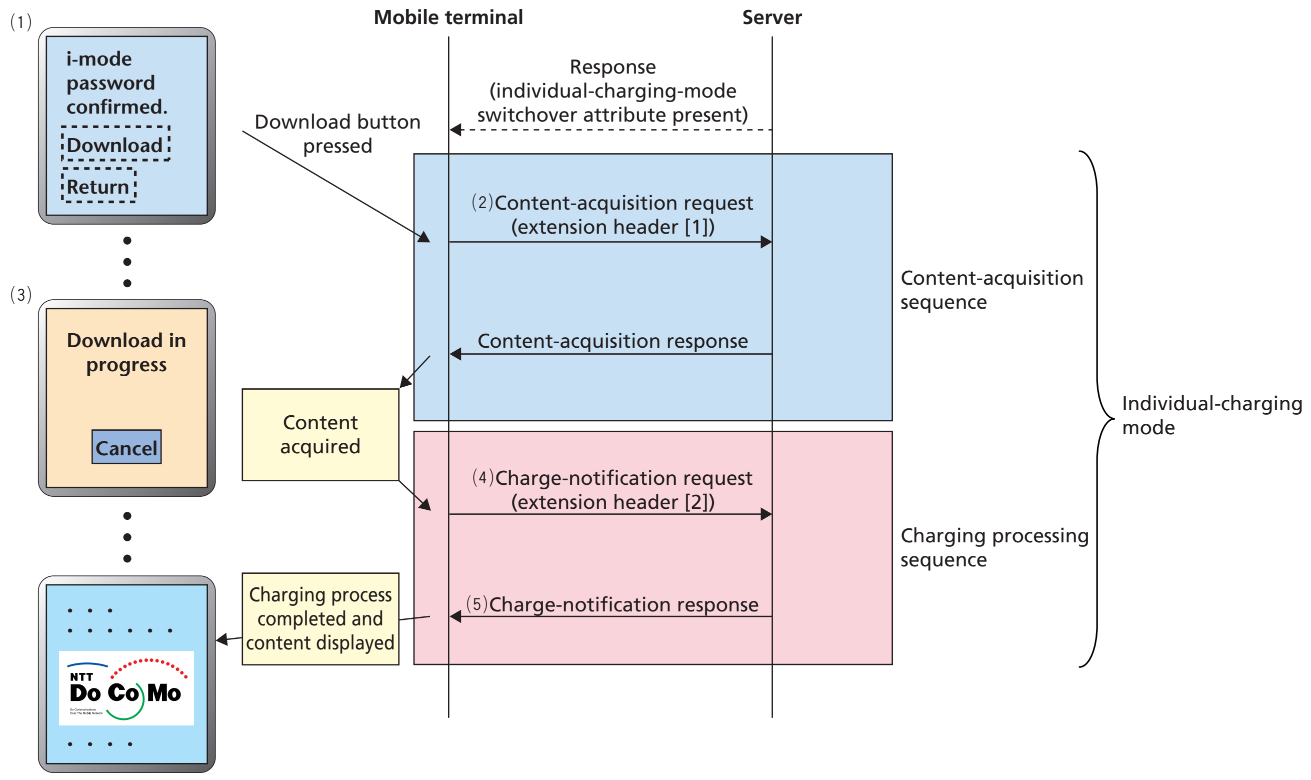

(1)
i-mode password confirmed.
Download
Return
(3)
Download in progress
Cancel
NTT DoCoMo
Mobile terminal
Server
Response (individual-charging-mode switchover attribute present)
Download button pressed
(2)Content-acquisition request (extension header [1])
Content-acquisition response
Content acquired
(4)Charge-notification request (extension header [2])
(5)Charge-notification response
Charging process completed and content displayed
Content-acquisition sequence
Charging processing sequence
Individual-charging mode

Figure 5 Basic sequence and screen transitions for individual charging

4) The charging process begins once content has been acquired. In particular, the mobile terminal, on recognizing that downloading has completed, accesses that content at an automatically determined internal URL and notifies the server side that the content in question has been downloaded (Fig. 5 (4)). Then, on the basis of this notification, the server returns a response to the terminal side and proceeds to execute the charging process for this user.
5) After receiving the charge-notification response (Fig. 5 (5)), the mobile terminal displays the target content.

5.3 Charge-notification Resend Function

Since mobile terminals use wireless communications, there is always the possibility that problems in communications will interrupt the data downloading. Such an interruption would result in the deletion of all data that has so far been received, a communication fee (packet volume) might be generated, and a disadvantage for the user. And there might also be a situation that the contents are all downloaded and ready but not able to display the contents, since the charge-notification processing has not completed. For this situation, we have designed a function that resends the charge notification only in the event that charge processing has failed in individual-charging mode thereby reducing disadvantageous situations for the user.

6. Conclusion

Against the background of equipping mobile terminals with a diverse functions, this article described the development of functions to improve usability by enabling the capture of content via FeliCa and making the charging system more flexible, and functions for improving user security by extending content protection and secure communications.

In the years to come, we expect the integration of terminal functions into the life infrastructure to accelerate and the requirements of security functions to become all the more severe. To cope with an ever increasing variety of usage formats for the ToruCa and terminal security functions, we plan to improve their usability and make their viewers even easier to use.

ABBREVIATIONS

CP: Contents Provider
CPRM: Content Protection for Recordable Media
HTML: Hyper Text Markup Language
HTTP: Hyper Text Transfer Protocol
LED: Light Emitting Diode
RootCA: Root Certificate Authority
SD: Secure Digital
SSL: Secure Sockets Layer
UIM: User Identity Module
URL: Uniform Resource Locator