Security Scan System Using Wireless Communication

Hironori Kobayashi, Takeshi Hayashi,

Koichi Asano and Masanori Fujita

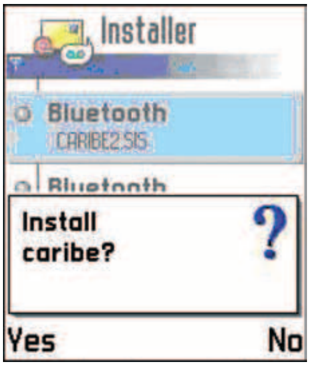
We have developed Security Scan System as a countermeasure against defects and malicious attacks in mobile terminals. The system allows detection of viruses and other problems in mobile terminals and update of pattern data via remote download using wireless communication.

1. Introduction

Viruses first appeared on the PC platform as programs that would destroy data, spreading via external storage media such as floppy disks. Later, other forms of viruses appeared due to the diffusion of the Internet and became a social problem. For example, self-replicating worms have spread explosively as they proliferate throughout networks via e-mails etc. In addition, leakage of personal and corporate information has become serious social issues in recent years. On the other hand, mobile terminals are becoming more and more sophisticated, as various advanced functions, such as electronic payment functions, are incorporated into the terminals, and they will without doubt become future targets of attacks. It is thus necessary to prepare for the following risks.

1) Intentional and malicious exploitation causing malfunctions as software program gets increasingly sophisticated.
2) Increased risks and greater damage due to diversification of intrusion paths.
3) Increased possibilities of attacks due to public disclosure of specifications resulting from usage of common mobile terminal platforms and conformance to standards.

At the time of writing this article (January 2005), new viruses targeting mobile terminals such as “Cabir” and “Skulls” are appearing (**Photos 1** and **2**). Recognizing such viruses and hacking as serious problems, researchers from all over the world are examining actions to be taken to protect network safety and reliability [1]. In order to defend mobile terminals against the risks as mentioned and maintain and improve the reliability of services provided, we have developed a system that updates pattern data for detecting these threats using a scan engine and wireless communication.



Install caribe?

Photo 1 “Cabir” screen image

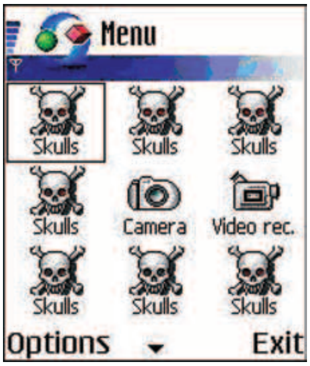


Photo 2 “Skulls” screen image

This system is for customers who use i-mode service and DoCoMo’s Internet Service Provider (ISP) connection service (a service where the user connects to the ISP via a packet network from a mobile terminal supporting i-mode) and is available on Freedom Of Mobile multimedia Access (FOMA) 901i series models, which were released in December 2004 and later.

This article presents an overview of the Security Scan System, and explains the technologies implemented in the mobile terminals supporting the scan engine and the server managing the pattern data, as well as the operation of the mobile terminals.

2. System Overview

Figure 1 shows an overview of the system. DoCoMo collects information on security risks related to mobile terminal software on a steady basis. If a problem is detected, pattern data for protecting the mobile terminals is created and entered into a mobile terminal software management server system, called the Mobile terminal Software Remote distributing system (MSR) (Fig. 1 (1) (2)). When a customer accesses the MSR from his/her mobile terminal (Fig. 1 (3)), pattern data optimized for that mobile terminal is downloaded only if the pattern data

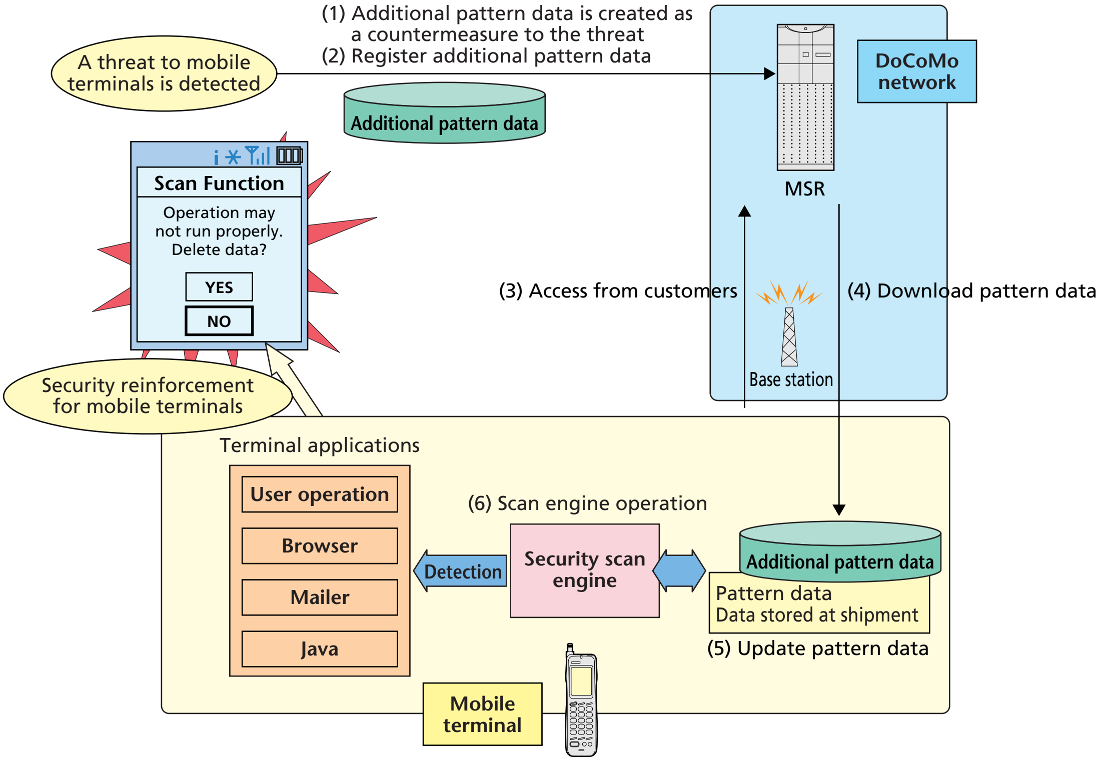

(1) Additional pattern data is created as a countermeasure to the threat
(2) Register additional pattern data
A threat to mobile terminals is detected
Additional pattern data
DoCoMo network
MSR
Scan Function
Operation may not run properly. Delete data?
YES
NO
Security reinforcement for mobile terminals
(3) Access from customers
(4) Download pattern data
Base station
Terminal applications
User operation
Browser
Mailer
Java
(6) Scan engine operation
Detection
Security scan engine
Additional pattern data
Pattern data
Data stored at shipment
(5) Update pattern data
Mobile terminal

Figure 1 Overview of Security Scan System

needs to be updated (Fig. 1 (4)). The details of the mobile terminal operation are discussed in Chapter 5. The downloaded pattern data is stored in the memory of the mobile terminal (Fig. 1 (5)). The scan engine in the mobile terminal refers to this pattern data and compares it with data used in various applications in the mobile terminal, with contents being accessed and with character strings in Uniform Resource Locators (URL) of sites in order to check in advance whether or not there are any matches. As a result, it became possible to prevent the use of data that may cause malfunctions and/or prohibit access to content or sites that execute malicious attacks before they cause any discomfort to the users (Fig. 1 (6)).

The system was made possible by downsizing the scan engine, which was based on detection technologies available for PCs, so that it could be embedded into mobile terminals. The result allows mobile terminals to conduct self-inspection of presence of viruses and other problems. The pattern data used by the scan engine is managed by the MSR servers deployed within the DoCoMo network. Whenever a new threat against the mobile terminals is discovered, users can avoid such a threat through remote download of latest pattern data released to counter the new threat. With this system, the security of the mobile terminals is reinforced in a timely and prompt manner.

Moreover, in order to shorten the system development period and reduce the cost, the servers that maintain pattern data were implemented based on the application and server facilities of the MSR developed in "Software Update System" [2].

3. Scan Engine Memory Optimization Technologies

The following two conditions are prerequisites for embedding a scan engine in mobile terminals.

1) Scan

The scan engine must be able to scan data handled by various applications using the pattern data in order to check and determine whether or not there are any viruses or other factors that may cause malfunctions.

2) Patten Data Update

The scan engine must be able to download pattern data using an interface between the mobile terminal and the MSR, extract the pattern data on the mobile terminal and store it in a storage medium or in memory.

Currently, the memory size required by a scan engine for PCs is in the order of several megabytes, but when implementing a scan engine for mobile terminals, it is necessary to take both the impact on the hardware and on the cost into consideration. Therefore, we decided to reduce the memory size required for the scan engine for mobile terminals to 1/10 of the size required for the PC. In preparation for this memory reduction of the scan engine, we examined how to reduce both the program code size and data size so that the conditions above could be satisfied.

First of all, if any of the basic scan functions were eliminated, the originally required security would likely be lowered. We thus decided not to eliminate the basic scan functions, but to eliminate only those that are not essential for mobile terminals, thereby optimizing the required memory size.

The pattern data update function is a function that is not frequently used. For this reason, all the memory used for the pattern data update processing is released after the completion of updating, thereby preventing the increase in the memory size.

The first specific example that was eliminated was the program code of the scan engine itself.

In order to do so, we clarified which parts of the existing code could be shared with the scan function, e.g., code for compressing and decompressing various files installed in mobile terminals and character code conversion code, and which parts that could not. In addition, we eliminated parts that were not necessary for the basic functions among the code that could not be shared. By eliminating codes that were not essential for mobile terminals, we were able to reduce a significant amount of program code related to the scan function.

Next, we reduced the amount of pattern data for scanning, which is transmitted to the mobile terminals from the MSR. In order to avoid large traffic increases and keep the impact on the current DoCoMo network to a minimum, we limited the size of one downloaded pattern data to approximately the size of one i-mode mail.

4. Pattern Data Download Traffic Control

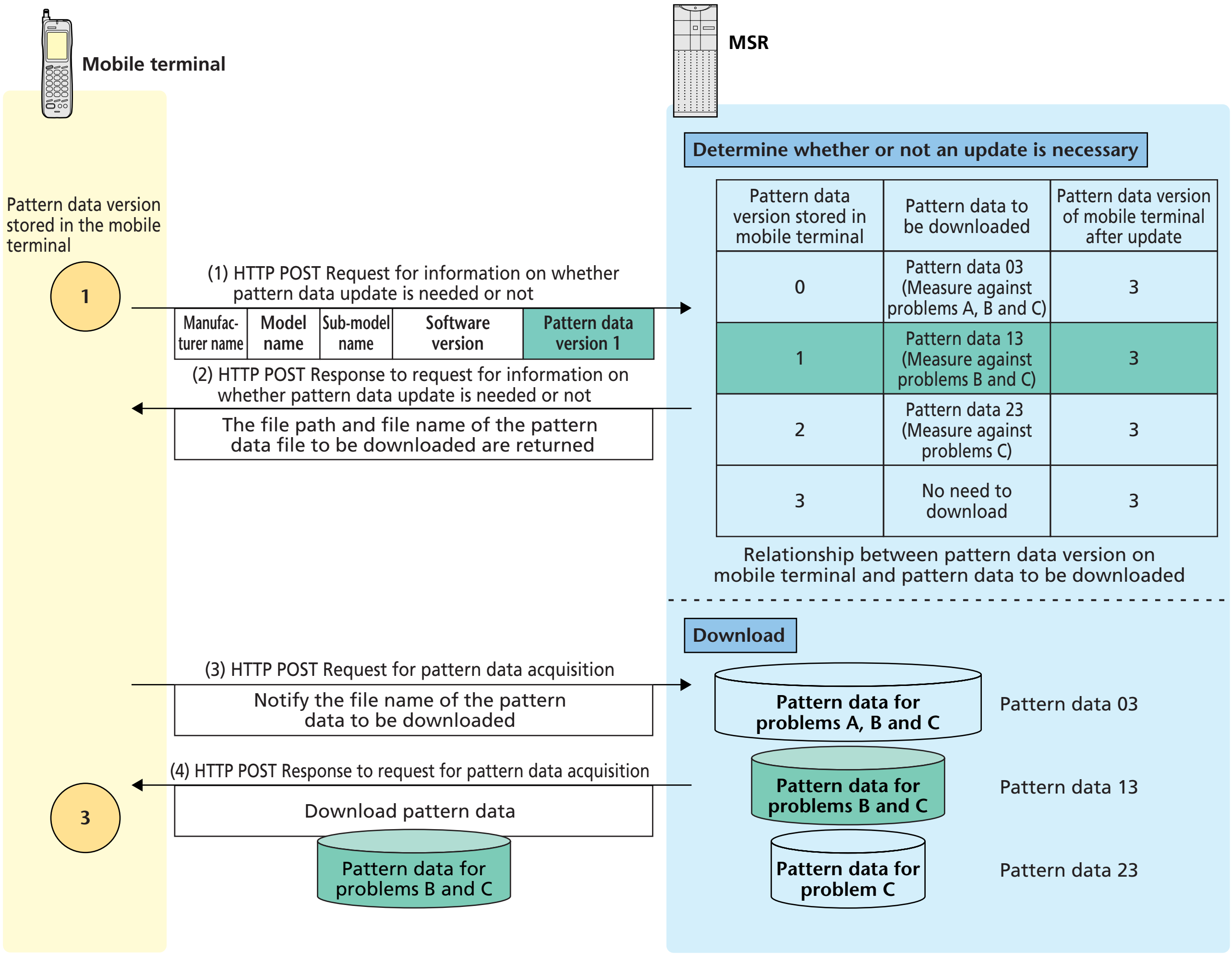
We analyzed and designed the management of the pattern data to be downloaded as follows, in order to suppress traffic increases on the DoCoMo network caused by the operation of this system and minimize the memory usage in the mobile terminals.

In the PC version of the scan engine, the same pattern data is provided without variations. In this system, we reduced the download data size by managing the pattern data per manufacturer, per model and per software version of each mobile terminal. When a problem unique to a certain version occurs, managing pattern data per software version suppresses traffic increases by limiting the pattern data downloads to only the version in question. Moreover, pattern data is created and managed exclusively for each problem type and software version of mobile terminals, thereby making it possible to download only differential updates to the pattern data that has already been downloaded, thus reducing the amount of data downloaded.

Next, the method of downloading pattern data is explained. The communication protocol used to download pattern data between mobile terminals and the MSR is HyperText Transfer Protocol (HTTP). The traffic is kept to a minimum by communicating only the essential data. Moreover, since HTTP is already proven to be a well-functioning protocol in the Software Update System, no changes are needed for the existing network; new functions are only added to the MSR and mobile terminals. HTTP POST method is used for communication of information between the mobile terminals and the MSR. As shown in **Figure 2**, pattern data is updated according to the following procedure: the manufacturer name, model name, sub-model name, software version and pattern data version information are entered into the message body of a HTTP POST request via mobile terminal operations and a query is sent to the MSR to check whether or not a pattern data update is needed (Fig. 2 (1)). If the pattern data needs to be updated, the file path and file name of the pattern data to be updated are placed in the message body of the HTTP POST response, which is then returned to the mobile terminal (Fig. 2 (2)). The mobile terminal then downloads the corresponding pattern data via HTTP POST (Fig. 2 (3) (4)).

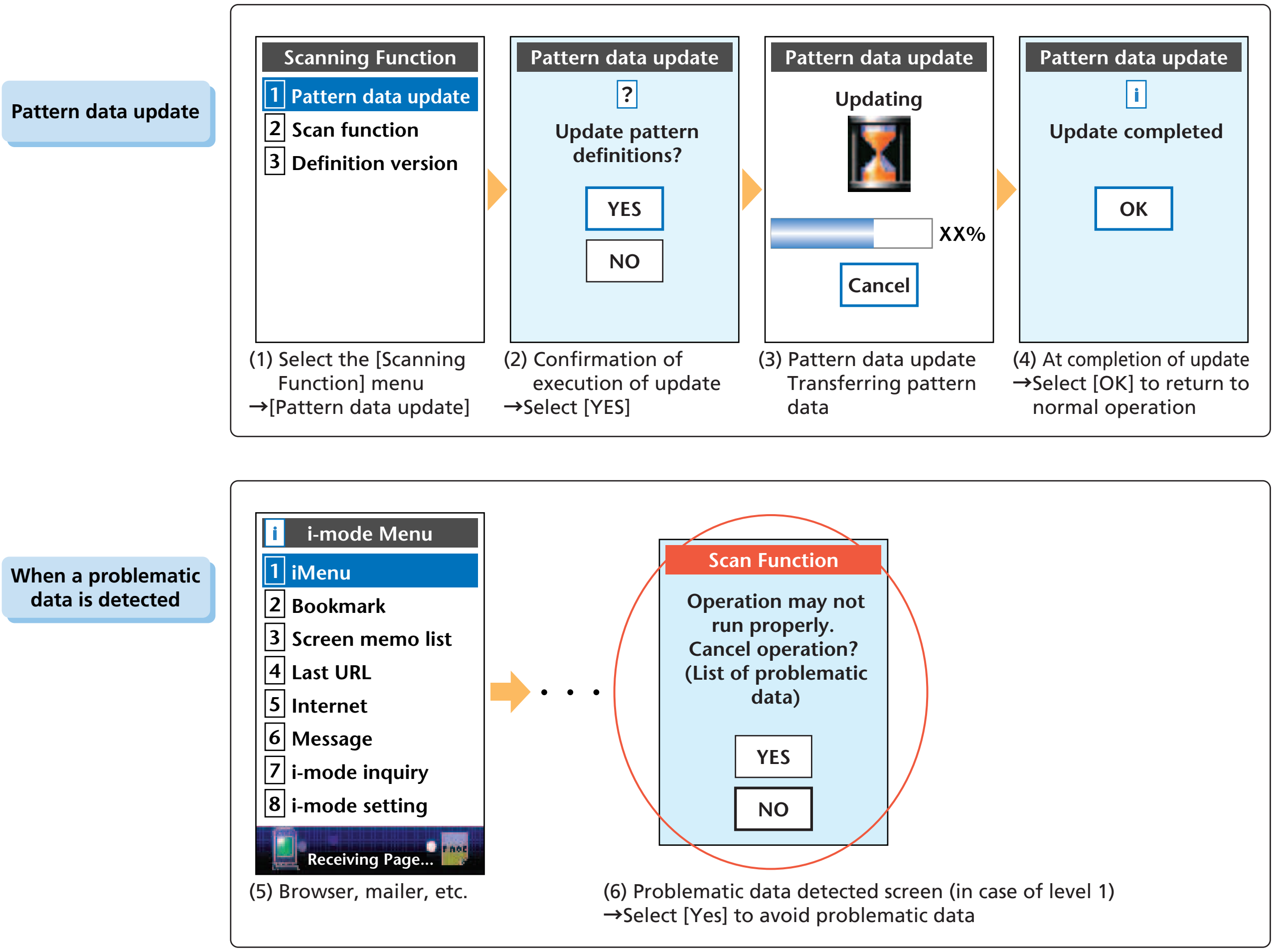
5. Mobile Terminal Operations

This chapter explains the operations involved in using the pattern data update and the scan function (**Figure 3**). Screens


Mobile terminal
MSR
Pattern data version stored in the mobile terminal
1
(1) HTTP POST Request for information on whether pattern data update is needed or not
Manufac-turer name
Model name
Sub-model name
Software version
Pattern data version 1
(2) HTTP POST Response to request for information on whether pattern data update is needed or not
The file path and file name of the pattern data file to be downloaded are returned
Determine whether or not an update is necessary
Relationship between pattern data version on mobile terminal and pattern data to be downloaded
Download
(3) HTTP POST Request for pattern data acquisition
Notify the file name of the pattern data to be downloaded
(4) HTTP POST Response to request for pattern data acquisition
Download pattern data
3
Pattern data for problems B and C
Pattern data for problems A, B and C
Pattern data 03
Pattern data for problems B and C
Pattern data 13
Pattern data for problem C
Pattern data 23

Pattern data version stored in mobile terminal	Pattern data to be downloaded	Pattern data version of mobile terminal after update
0	Pattern data 03 (Measure against problems A, B and C)	3
1	Pattern data 13 (Measure against problems B and C)	3
2	Pattern data 23 (Measure against problems C)	3
3	No need to download	3

Figure 2 Specification of interface between mobile terminal and MSR



Update pattern definitions?

Update completed

Operation may not run properly. Cancel operation? (List of problematic data)

(1) Select the [Scanning Function] menu →[Pattern data update]

(2) Confirmation of execution of update →Select [YES]

(3) Pattern data update Transferring pattern data

(4) At completion of update →Select [OK] to return to normal operation

(5) Browser, mailer, etc.

(6) Problematic data detected screen (in case of level 1) →Select [Yes] to avoid problematic data

Figure 3 Mobile terminal screen images at pattern data update and detection of problematic data

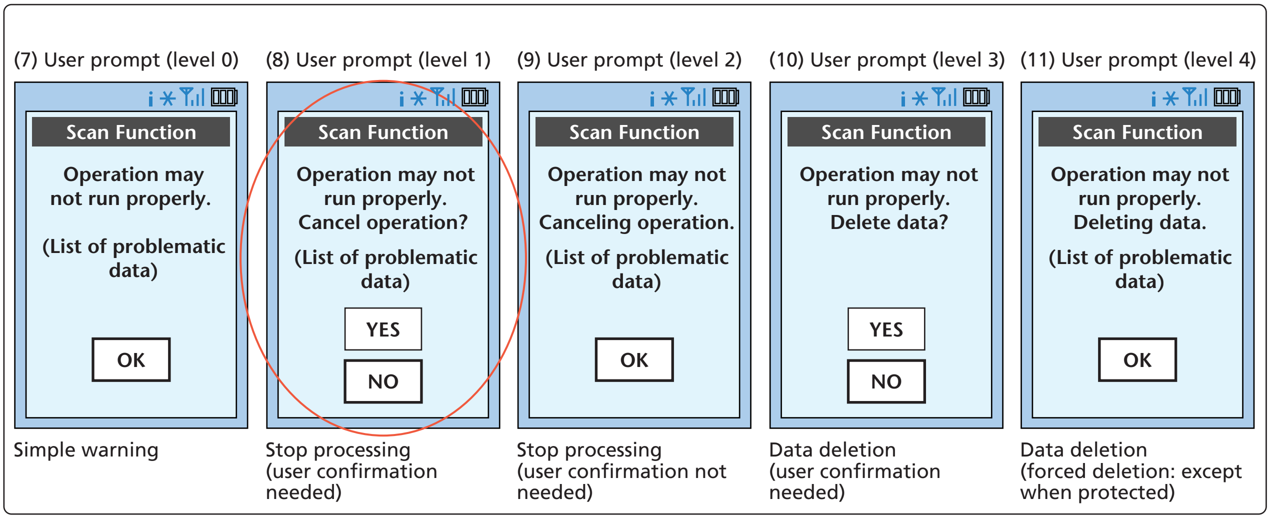
(1) to (4) show the operations involved in using the pattern update function. User can display screen (1) at any desired point, which is the configuration menu of the scan function (the exact position of the menu varies depending on the mobile terminal model). To update the pattern data, select [Pattern data update]. The pattern data update confirmation screen (2) appears; select [YES] to start downloading (screen (3)). When the pattern data update is completed, screen (4) appears and the update is completed.

The scan result screen, on the other hand, appears when a problematic data is detected; the display timing varies depending on the situation. For example, screens (5) and (6) may appear if a problem was detected while viewing a certain site selected from the i-mode menu in a browser. Screen (6) appears if the scan function detects information that may cause a problem. It should be noted that pattern data contains information on what actions are available to the user after a screen is displayed to notify that problematic data has been discovered. This is because we considered it necessary to allow a certain amount of control on the mobile terminal side as well because it is not possible to perform a single countermeasure that can deal with problems of all levels and also because mobile terminal users cannot be expected to be familiar with viruses as much as PC users. The problem level values are stored in the pattern data together with the characteristics of problematic data. Screen (6) is an example where the level value is 1. Screens (7) to (11) of **Figure 4** show examples of other levels.

In the case of pattern data for scanning of level 1, a user can select whether or not to display the problematic data. In case of attacks by actual viruses, it is not always a good idea to allow the user to decide the action to take, however. In this case, it is possible to carry out various control procedures on the code, from prohibiting access to forced deletion of data, as shown in screens (9) to (11). This function was also newly designed during the development.

6. Conclusion

This article described Security Scan System that updates



(7) User prompt (level 0)

Scan Function

Operation may not run properly.

(List of problematic data)

Simple warning

(8) User prompt (level 1)

Scan Function

Operation may not run properly. Cancel operation?

(List of problematic data)

Stop processing (user confirmation needed)

(9) User prompt (level 2)

Scan Function

Operation may not run properly. Canceling operation.

(List of problematic data)

Stop processing (user confirmation not needed)

(10) User prompt (level 3)

Scan Function

Operation may not run properly. Delete data?

Data deletion (user confirmation needed)

(11) User prompt (level 4)

Scan Function

Operation may not run properly. Deleting data.

(List of problematic data)

Data deletion (forced deletion: except when protected)

* It is possible to display user interfaces at different levels depending on the problem detected

Figure 4 Screen images at problem detection

pattern data by remote download using wireless communication, which is based on the scan engine technology for mobile terminals and the pattern data download technology of the MSR server. This function, which was first introduced on FOMA 901 series models, will be available on forthcoming FOMA models as well. In the future, we intend to examine other possible applications of Security Scan System and its implementation technologies.

References

[1] Ministry of Internal Affairs and Communications: "Study Group on Safety and Reliability of 3G Mobile Communication Systems," Technical Report, 2001 (in Japanese).

[2] S. Hoshi et al.: "Software Update System Using Wireless Communication," NTT DoCoMo Technical Journal, Vol. 11, No. 4, pp. 36–41, Jan. 2004.

Abbreviations

FOMA: Freedom Of Mobile multimedia Access
HTTP: HyperText Transfer Protocol
ISP: Internet Service Provider
MSR: Mobile terminal Software Remote distributing system
URL: Uniform Resource Locator