Proceedings of the 2004 IEEE
International Conference on Networking, Sensing & Control
Taipei, Taiwan, March 21-23, 2004

# A Remote Control Scheme for Ubiquitous Personal Computing

**Pan-Lung Tsai**
Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan
charles@fractal.ee.ntu.edu.tw

**Chin-Laung Lei**
Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan
lei@cc.ee.ntu.edu.tw

**Wen-Yang Wang**
Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan
wwj@fractal.ee.ntu.edu.tw

**Abstract** – *In the schemes designed for ubiquitous computing, remote access to private data is generally achieved by requiring users to deposit the data on certain always-on-line servers in the network infrastructure. However, when the data to be accessed remotely is somewhat sensitive, as implied by many applications of personal computing, users may prefer to warehouse the data in their residences rather than in some public places. In this paper, a novel scheme is proposed to realize personal computing ubiquitously by satisfying such demand. The proposed scheme employs home-automation techniques to grant properly authenticated users remote access to the data stored somewhere off the infrastructure (e.g., their personal workstations) while avoiding the expense of deploying high-availability servers at home. It also provides three kinds of authentication mechanisms and hence allows users to choose among different levels of the tradeoff between security and ubiquity, providing a complete solution for ubiquitous personal computing.*

**Keywords:** Ubiquitous computing, personal computing, remote desktop control, home automation, authentication.

## 1 Introduction

The primary goal of ubiquitous computing is to allow users to perform various computations anywhere. Under current development of the Internet, this goal can be satisfied by the combination of the extensively deployed Web browsers and a wide variety of Web services. On one hand, Web services are able to provide plenty of applications. On the other hand, Web browsers enable users to make use of these applications virtually from any location on earth.

The Web-based solutions mentioned above are perfect except for those computational tasks that involve processing of private data, as in many applications of personal computing. Insisting on such solutions will then require users to deposit their private data to the providers of the publicly available Web services, or alternatively they will have to deploy and maintain their own Web services. The former approach results in many security issues, while the other one incurs prohibitive expenses to most users. Applications falling into the overlapped area of ubiquitous computing and personal computing deserve more elegant solutions.

Recent development of home automation techniques makes possible remote access to household information appliances across the Internet. Although these techniques may serve as fundamental mechanisms of remote control, they do not offer as many applications that help users remotely access information as Web services do. As a complement, the technology of remote desktop control can be adopted to allow users to operate their personal workstations remotely. Since users can perform various operations exactly in the same way as how they accomplish these tasks at home, the application support is even better than that of Web services. Besides, these personal workstations are not required to always stay on-line. In fact, the power of the workstations can be switched off when not in use.

In this paper, a remote control scheme is proposed as a solution for those applications falling into the intersection of ubiquitous computing and personal computing. The rest of the paper is organized as follows. Section 2 first describes the technologies used, followed by detailed explanation of the proposed scheme in section 3. Section 4 then illustrates three scenarios where the proposed scheme is applicable with different levels of the tradeoff between security and ubiquity. Section 5 compares the proposed scheme to related works, and section 6 summarizes our achievements.

## 2 Enabling technologies

In the design and implementation of the proposed scheme, the most important enabling technologies are residential gateways, power management for personal computers, remote desktop control, and GSM (Global System for Mobile communication) SMS (Short Message Service). In this section, residential gateways and power management for personal computers are described briefly. For the details of remote desktop control and the use of GSM SMS as an authentication mechanism, please refer to our earlier work in [10].

### 2.1 Residential gateways

In the past, only big organizations and enterprises are able to afford the deploying and maintenance cost of

*This research was supported in part by the National Science Council of the Republic of China under grant NSC-92-2213-E-002-012.*

broadband connectivity. However, the advancement of last-mile broadband technologies such as DSL (Digital Subscriber Line) and CMTS (Cable Modem Termination System) has resulted in radical changes and successfully moved the boundaries of the Internet from service providers to customer premises. Nowadays, the borders of the so-called infrastructure comprise a wide variety of edge devices residing in houses.

In addition to the provision of connectivity to the rest of the world, such edge devices have evolved into residential gateways [6] and become a control and management point of information appliances in the household [8]. Similar to other network-centric components, residential gateways typically communicate with other devices via IP (Internet Protocol) over different underlying communication media (e.g., infrared links, coaxial cables, twisted-pair wires, etc.). Modern residential gateways also incorporate extra functions like packet filtering, network address and port translation, or even handling basic-level HTTP (HyperText Transfer Protocol) client requests.

### 2.2 Power management for personal computers

The standard ACPI (Advanced Configuration and Power Interface) specification [3] defined several approaches of waking up a personal computer remotely. A widely adopted implementation of turning on remote computer via specially patterned network packets is Magic Packet™ technology [1], first introduced by Advanced Micro Devices, Inc. in 1995. In the present time, this implementation is best known by its synonym, WOL (Wake On LAN, Local Area Network).

Another feature relevant to turning personal computers into information appliances is the definitions of five sleeping states. Each sleeping state bears a different degree of power consumption and wake-up latency. For personal computers complying with ACPI standard, users need not to turn off the power every time they finish the jobs on their personal computers. Instead, the computers can be configured to automatically enter specified sleeping states after certain period of idle time. The sleeping state S4, also known as the hibernation state, which consumes the least amount (the same as normally turned-off computers) of electric power, is especially useful because it requires the software contexts to be saved before computers go into hibernation. This special requirement results in shorter start-up time on next boot of the hibernated computers, making the use of personal computers closer to commodity electronic appliances.

## 3 The proposed scheme

Imagine that freelancer Jason has taken a twelve-day trip abroad on his vacation. Unfortunately, when Jason returns to the hotel on the fourth day, he gets a message left by one of his clients, asking him to handle an urgent case. If Jason is currently sitting in front of his personal workstation at home, the request from the client is actually a piece of cake. Now Jason is struggling about whether he should give up the remaining schedule and fly home immediately. This is a typical scenario in which the proposed scheme comes to rescue. With the help of our scheme, Jason may stroll to one of the publicly available Web terminals in the neighborhood and solve the problem in time, instead of rushing to the nearest airport and taking the next flight home.

The proposed scheme is composed of five consecutive phases, which are elaborated in subsection 3.1. Subsection 3.2 explains our implementation of a system prototype.

### 3.1 Procedure of the proposed scheme

As shown in Figure 1, the realization of ubiquitous personal computing using the proposed scheme goes through the following phases in sequence.

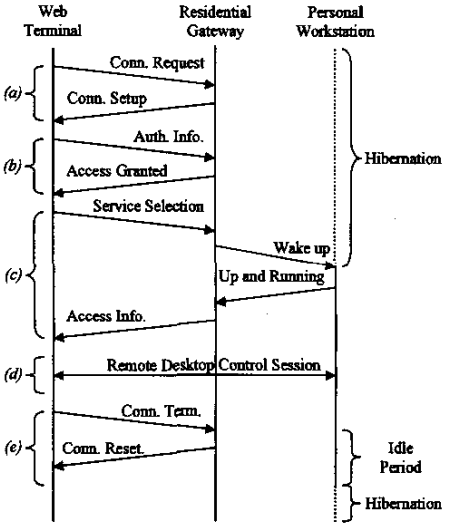


Figure 1. Procedural diagram of the proposed scheme

*(a)* The signaling phase: This is the very first phase of the whole procedure. In the signaling phase, the user attempts to establish a conceptually exclusive communication channel between the terminal and his residential gateway at home.

*(b)* The authentication phase: After the exclusive communication channel has been set up, the user is required to prove his identity to the residential gateway via some predetermined authentication mechanisms.

*(c)* The activation phase: The successfully authenticated user then asks the residential gateway to turn on his personal workstation on his behalf. If there are two or more workstations in the house, the user also has to specify which one to turn on. To turn on the specified workstation, the residential gateway makes use of the techniques mentioned in subsection 2.2.

*(d)* The service phase: Upon entering this phase, the user runs a Web-based client for remote desktop control

to operate his personal workstation remotely. The adoption of remote desktop control software enables the user to perform various operations exactly in the same way as how he accomplishes these tasks when sitting in front of his workstation at home.

*(e)* The completion phase: When the user finishes his work, he may actively shutdown the personal workstation before breaking the communication channel between the terminal and the residential gateway. Alternatively, if the user simply breaks the communication channel without turning off the machine, the workstation will then automatically enter the hibernation state after a predetermined period of idle time.

Note that the five phases mentioned above do not imply five separate actions carried out by the user in each session. In section 4, we will see that two or more phases may be combined in a single action.

### 3.2 Implementation of the system prototype

The implementation of the proposed scheme involves the creation of a specialized residential gateway and proper configurations of the personal workstation. In order to understand the complexity of the implementation and to provide the proof of concept, we have constructed a system prototype. Figure 2 shows the working model of the system prototype.

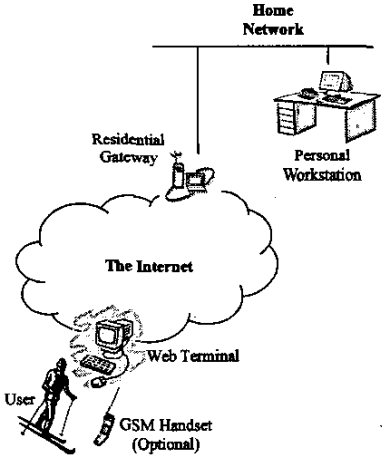


Figure 2. System model

In the system prototype, we use a personal computer with a WOL–capable network interface card to serve as the personal workstation to be controlled remotely, and another personal computer with two network interface cards and a GSM modem (in the form of an add-on card) to implement the residential gateway. The personal workstation runs Windows® 2000 Professional operating system, and has WinVNC and Java VNC Viewer (remote desktop control software based on VNC, Virtual Network Computing [7]) installed. We choose WinVNC and Java VNC Viewer because of their wide acceptance in related researches. Since we do not rely on any particular characteristics of the underlying remote desktop control protocol and software, they can be substituted by other implementations of remote desktop protocol (e.g., Microsoft® Windows® NetMeeting® 3 and NetMeeting® UI ActiveX® Control, respectively) as needed. The residential gateway also runs Windows® 2000 Professional operating system and uses the built-in ICS/ICF (Internet Connection Sharing/Internet Connection Firewall) feature to provide NAT/NAPT (Network Address Translator/Network Address Port Translator) [9] and firewall functions. Besides, the Web server on the residential gateway is enabled to allow Web-based authentication (which will be explained later). A third personal computer running Windows® 98 operating system is used as the Web terminal. In fact, the terminal can be any much less powerful device, as long as it runs Web browsers.

The residential gateway always stays on-line, while the personal workstation is only turned on when needed. The personal workstation is protected by the firewall function of the residential gateway and normally cannot be reached from the Internet, even when the power is turned on. That is, though the user can use his personal workstation to connect to the Internet by setting up outbound network connections, connection requests in the reverse direction will be completely denied. When it is necessary to access the personal workstation from outside world, the settings on the residential gateway must be temporarily changed to allow selective inbound network connections to the personal workstation.

The authentication process in our system prototype is accomplished by sending user account information and the corresponding password to the residential gateway via an exclusive communication channel. Currently we allow the user to establish three kinds of channels: Web connections based on SSL (Secure Socket Layer), GSM SMS, and PSTN (Public Switched Telephone Network) circuits. The actual steps taken by the user and the security considerations when using different kind of channels are discussed in section 4. Here we concentrate on the software architecture of the residential gateway.

Figure 3 shows the software components of the residential gateway. Component *WebAuth* is a collection of HTML (HyperText Markup Language) pages responsible for displaying a login prompt when the user types the URL (Uniform Resource Locator) associated with the residential gateway in a Web browser. It is also responsible for notifying the *WakeWS* component upon a successful authentication as well as displaying the result of authentication back to the user. Component *SMSAuth* has similar functionality to *WebAuth*, except that it receives and transmits data via GSM SMS messages rather than SSL connections. Component *VoiceAuth* is basically an IVR (Interactive Voice Response) system that responds to incoming phone calls by playing a set of previously recorded voice messages in reaction to the DTMF (Dual

Tone Multi-Frequency) signals generated by pressing the touch-tone buttons on the telephone. It is implemented using TAPI (Telephony Application Programming Interface) and is a little bit more complicated than the other two. *SMSAuth* and *VoiceAuth* are integrated in a single Win32® service, where *SMSAuth* keeps monitoring incoming SMS messages and *VoiceAuth* constantly waits for incoming phone calls.

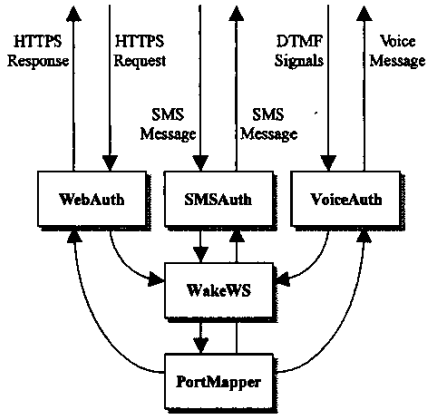


Figure 3. Interactions between software components

The primary task of component *WakeWS* is to turn on the specified personal workstation when it gets notification of a successful authentication from one of *WebAuth*, *SMSAuth*, and *VoiceAuth*. Upon receiving the name of the personal workstation to be controlled remotely, *WakeWS* first looks up the corresponding IP address and MAC (Media Access Control) address of the personal workstation from an internal table. Then it formulates a specially patterned Ethernet packet (the magic packet) accordingly and broadcasts it over the in-house network. After a short period of waiting, *WakeWS* determines the status of the personal workstation by sending a couple of ICMP (Internet Control Message Protocol) Echo messages to it and checks whether there is any reply. If the personal workstation of interest is successfully turned on, as implied by the existence of ICMP Echo Reply messages, *WakeWS* then asks component *PortMapper* to create one or more temporary port-mapping entries in the NAT/NAPT table by means of ICS/ICF API (Application Programming Interface) so that selective inbound network connections to the personal workstation can be established. Afterwards, *PortMapper* notifies the originating authentication component (one of *WebAuth*, *SMSAuth*, and *VoiceAuth*) to send the acknowledgement back to the user through the same communication channel.

By setting up corresponding port-mapping entries in the NAT/NAPT table, the residential gateway restricts permissible inbound connections only to those originated from the IP address of the Web terminal that the authenticated user currently uses. Using one-time external port also helps enhance the security. When the user gets the response, he extracts the URL embedded and types it in a Web browser. Then the Java VNC Viewer installed on his personal workstation at home will be downloaded to the terminal. Now he can see the desktop of his personal workstation and operate the machine.

During the remote desktop control session, the work involved by the residential gateway is minimized. The residential gateway is only responsible for translating addresses and ports in the packets resulted from the running session. Such tasks are normally accomplished by NAT/NAPT software logics at layer 3 and layer 4. In most implementations, the computation power required to perform NAT/NAPT translations on a small number of TCP connections is trivial.

When the inbound connections to the personal workstation are broken, the corresponding port-mapping entries will expire and eventually be deleted by NAT/NAPT garbage collector. As mentioned earlier, if the personal workstation is not turned off before the disconnection, it will enter the hibernation state automatically after a predetermined period of idle time. In our implementation, the personal workstation is configured to hibernate after fifteen minutes of idle period. When the personal workstation is turned off or goes into hibernation, the whole system returns to the original starting state.

# 4 Tradeoff between security and ubiquity

Recall Jason we mentioned in section 3. Here we are going to describe three different scenarios in which Jason carries different devices during the trip as well as the actual steps taken by Jason to remotely control his personal workstation at home. In subsection 4.1, workaholic Jason brings a powerful laptop with him during the trip. In subsection 4.2, Jason only has his GSM handset beside him. In subsection 4.3, Jason does not have any personal computation and communication devices. The security levels of the scenarios are also discussed respectively. When discussing security issues here, we merely focus on the possible leakage of authentication information. Other security issues, such as securing the transmission of the data in the remote desktop control session, or securing the final display on the terminal, are not considered in this paper. The three scenarios together clearly illustrate the idea of the tradeoff between security and ubiquity.

## 4.1 Using private terminals

If the user in a remote location has his own private terminal such as a laptop, he will take the following steps to access his personal workstation at home.

(1) The user runs a Web browser on his private terminal and types the URL associated with his residential gateway (e.g., https://myhome.dyndns.org/). The browser then shows the initial login page.

(2) The user types his account, password, the name of the personal workstation to be controlled remotely, and current IP address of the private terminal in the login page, and submits these items back to the residential gateway. Upon receiving the data, the residential gateway performs validation on the submitted information and turns on the specified personal workstation if the authentication succeeds. After the personal workstation is turned on, the residential gateway sets up temporary NAT/NAPT port-mapping entries as mentioned in subsection 3.2, and returns a dynamically generated HTML page containing the corresponding URL (e.g., http://myhome.dyndns.org:2363/) to the user.

(3) The user then types the received URL and the browser shows the initial screen of the Java VNC Viewer. Now the user is ready to use the remote desktop control software to operate his personal workstation remotely.

(4) When the user finishes his work, he terminates the Web browser. His personal workstation will then go into hibernation in fifteen minutes.

In the description above, steps (1), (3), and (4) corresponds to phases *(a)*, *(d)*, and *(e)* mentioned in subsection 3.1, respectively, while step (2) combines the authentication phase, *(b)*, and the activation phase, *(c)*.

The scenario described in this subsection is actually a typical situation one may encounter in the field of mobile computing. Since both sides (the private terminal and the residential gateway) of the communication channel are trusted parties under the control of the same authority, it is relatively easy to implement any necessary security mechanism to keep the authentication information unexposed. However, such highest level of security is achieved at the price of inconvenience, or less degree of ubiquity. Strictly speaking, requiring a traveling user to carry his private terminal in fact makes this scenario fall into the domain of mobile computing rather than ubiquitous computing, as mentioned in the beginning of this paragraph.

### 4.2 Using GSM network with public terminals

When the user has a GSM handset in hand, as most people do when traveling, he may find a publicly available terminal and take the following steps.

(1) The user composes an SMS message by concatenating his account, password, the name of the personal workstation to be controlled remotely, and the IP address of the public terminal, using commas as separators. Then he sends the message to the phone number associated with his residential gateway. Upon receiving the message, the residential gateway parses it and performs validation on the authentication information. If nothing goes wrong, the residential gateway turns on the specified personal workstation, set up the temporary NAT/NAPT port-mapping entries, and sends back another SMS message containing the corresponding URL (e.g., http://myhome.dyndns.org:5251/).

(2) [This step is the same as step (3) in subsection 4.1.]

(3) [This step is the same as step (4) in subsection 4.1.]

In the description above, steps (2), and (3) corresponds to phases *(d)*, and *(e)* mentioned in subsection 3.1, respectively, while step (1) combines the signaling phase, *(a)*, the authentication phase, *(b)*, and the activation phase, *(c)*.

This scenario relieves the user from the necessity of carrying a computation device to serve as the private terminal when traveling and hence exhibits higher degree of ubiquity than the previous one. Owing to the wide acceptance and heavy utilization of cellular phones, taking around a GSM handset is seldom considered as an extra burden for most of us. We also argue that the procedure described in this subsection retains reasonable security since a secondary communication channel (with respect to the Internet), GSM network, is used as a secure channel to transmit the authentication information, and preventing eavesdropping is one of the primary goals in the design of GSM network [2].

### 4.3 Using PSTN network with public terminals

Even if the user does not bring his own computation and communication devices with him, he may still access his personal workstation at home by first finding a payphone and a publicly available terminal, and then carrying out the steps stated as follows.

(1) The user makes a phone call by dialing the phone number associated with his residential gateway. The residential gateway then answers the phone call by playing a previously recorded voice message.

(2) The user listens to the playback of the voice message and follows the instructions to enter his account and password (both encoded in alphanumeric symbols) by pressing the touch-tone buttons on the phone. When receiving this information, the residential gateway verifies its validity. If the information supplied by the user passes the authentication process successfully, the residential gateway continues to ask the user for further information by playing another previously recorded voice message.

(3) Then the user enters the name of the personal workstation to be controlled remotely and the IP address of the public terminal (both encoded in alphanumeric symbols) via the touch-tone buttons when asked by the playback of the voice message. Upon receiving the requested information, the residential gateway turns on the specified personal workstation and sets up the temporary NAT/NAPT port-mapping entries. Then the residential gateway dynamically composes a voice message indicating the corresponding URL (e.g., http://myhome.

dyndns.org:8247/) and plays it to the user through the phone line.

(4) [This step is the same as step (3) in subsection 4.1.]

(5) [This step is the same as step (4) in subsection 4.1.]

In the description above, each step corresponds to a single phase mentioned in subsection 3.1.

This scenario demonstrates how to take advantage of widely deployed facilities, PSTN network and Web terminals, to accomplish the goal of ubiquitous computing. The degree of ubiquity achieved is the highest among the three since the user is not required to carry any computation or communication device. However, the security is somehow compromised because PSTN network is generally not considered secure enough and the number of different DTMF tones is so small that they can be easily interpreted or remembered by an eavesdropper. A few common security techniques, such as challenge and response, may be adopted to reduce the chance of successful security attacks, but the restricted capabilities of ordinary telephones greatly limit the complexity of possible implementation and hence diminish the effectiveness of these techniques.

# 5 Related works

Some previous works, such as [4] and [5], also made the proposal of integrating specific remote desktop control software into residential gateways or information appliances. In spite of their pioneering contributions, there were insufficiencies in their works. The scheme in [4] depended on extensions to original VNC [7] protocol and hence limited itself to that particular remote desktop control software. The scheme appeared in [5] not only bound itself to VNC but also required the execution of an application-level proxy on somehow resource-limited devices in practice. Both of the works focused on the provision of remote desktop control mechanisms and did not consider the issues specific to ubiquitous computing and personal computing (e.g., authentication).

Compared to them, our scheme does not modify the underlying remote desktop control protocol and software, and therefore is not limited to particular remote desktop control implementation. In addition, most of the work during remote desktop control session is accomplished by NAT/NAPT software logics at layer 3 and layer 4 so that the computation overhead is greatly reduced, resulting in a light-weighted design of residential gateways. The proposal of using ACPI [3] to control the power states of personal workstations from residential gateways also makes the proposed scheme unique.

# 6 Conclusion

In this paper, a novel scheme is proposed to realize ubiquitous personal computing. The proposed scheme integrates home-automation techniques, the technology of remote desktop control, and different authentication mechanisms to form a total solution. More specifically, power management functions allow users to remotely access their personal data stored at home without the expense of deploying and maintaining highly available servers on their own, while remote desktop control enables users to remotely operate their personal workstations exactly in the same way as how they use the workstations when sitting in front of the consoles. In addition, the authentication mechanisms implemented on the residential gateway permit users to choose among different levels of the tradeoff between security and ubiquity.

# References


[1] Advanced Micro Devices, Inc., "Magic Packet Technology White Paper," Revision A, November 1995. http://www.amd.com/us-en/ConnectivitySolutions/TechnicalResources/0,,50_2334_2481_2494,00.html.

[2] C. Brookson, "GSM security: a description of the reasons for security and the techniques," Proceedings of IEE Colloquium on Security and Cryptography Applications to Radio Systems, pp. 2/1-2/4, June 1994.

[3] Compaq Computer Corporation, Intel Corporation, Microsoft Corporation, Phoenix Technologies, Ltd., and Toshiba Corporation, "Advanced Configuration and Power Interface Specification," Revision 2.0c, August 2003. http://www.acpi.info/spec.htm.

[4] T. Haraikawa, T. Sakamoto, T. Hase, T. Mizuno, and A. Togashi, "μVNC over PLC: a framework for GUI-based remote operation of home appliances through power-line communication," IEEE Transactions on Consumer Electronics, Vol. 48, No. 4, pp. 1067-1074, November 2002.

[5] A. Hasedawa and T. Nakajima, "A user interface system for home appliances with virtual network computing," Proceedings of 2001 IEEE International Conference on Distributed Computing Systems Workshops, pp. 229-234, April 2001.

[6] C. R. Holliday, "The residential gateway," IEEE Spectrum, Vol. 34, No. 5, pp. 29-31, May 1997.

[7] T. Richardson, Q. Stafford-Fraser, K. R. Wood, and A. Hopper, "Virtual network computing," IEEE Internet Computing, Vol. 2, No. 1, pp. 33-38, January 1998.

[8] T. Saito, I. Tomoda, Y. Takabatake, K. Teramoto, and K. Fujimoto, "Gateway technologies for home network and their implementations," Proceedings of 2001 IEEE International Conference on Distributed Computing Systems Workshops, pp. 175-180, April 2001.

[9] P. Srisuresh and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)," RFC 3022, January 2001. http://www.ietf.org/rfc/rfc3022.txt.

[10] P.-L. Tsai and C.-L. Lei, "Towards ubiquitous computing via secure desktop service," Proceedings of IEEE Region 10 International Conference on Electrical and Electronic Technology, 2001, Vol. 1, pp. 187-190, August 2001.