A Novel Mobile Content Delivery Scenario with Simple Double-Key Secure Access Control

Chih-Lin Hu and Chien-An Cho
Department of Communication Engineering
National Central University
Taoyuan, Taiwan, R.O.C.
clhu@ce.ncu.edu.tw; 955003020@cc.ncu.edu.tw

Abstract

Under the convergence of fixed and mobile network systems, modern networked devices are often equipped with multiple connectivity modules so that users can access various information services, by means of portable information devices, anytime and anywhere over ubiquitous Internet access connectivity. In this paper, we present a novel secure mobile content delivery mechanism where networked devices in a vicinity can discover each other, transfer media contents in a convenient, networked method instead of ordinary transfer method that involves unfriendly manual operations of connection setup and file transfer. Its design integrates several significant components, including device discovery, mobile content delivery and double-key secure access control, which are able to alleviate several inherent limitations in wireless and mobile networks. It thus enables mobile handheld devices to escape from mobility confinement and to transfer media contents in an efficient, energy-saving and secure manner.

1. Introduction

Current development of fixed and wireless network communication systems converges toward an integrated, hybrid network environment, as depicted in Figure 1. Many mobile handheld devices (MHDs)[1] are often equipped with multiple connectivity modules. A typical example is the sort of mobile phones that have 3G, Wi-Fi, and Bluetooth functions. An MHD can attach to multiple network contexts simultaneously in such an environment. On the other hand, modern MHDs are equipped with digital still camera, audio recording and audio video (AV) processing modules, with that users can take pictures, shoot video clips, or record audio sounds conveniently and effectively. Not only downloading Internet files and media contents, but also MHDs produce multimedia contents. With more and more external storage space, the MHDs can cumulatively store larger volume of media contents and then serve as mobile content servers. It is natural and usual, hence, users are willing to exchange or share media contents with others among different networked devices.

The work in this paper considers a novel secure mobile content delivery scenario in an integrated fixed, wireless and mobile network environment, as described in Scenario 1 below. Rather than many mobile information services that are dedicated to reproduce similar scenarios of information retrieval, currently in traditional, fixed networks, into wireless and mobile network environments, our thinking is to exploit new potential user scenarios, possibly introduced by the advance of wireless communication technologies and the MHD's capabilities, thus distinguishing our work from the others.

[1] We use this term hereinafter which implicitly represents all sorts of wireless networked devices in an integrated, hybrid network environment.

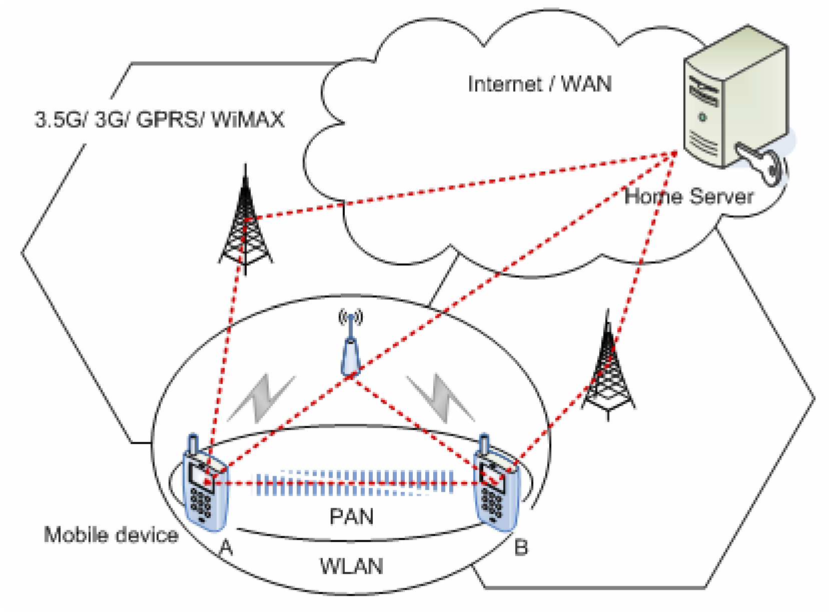

Figure 1. A hybrid network environment

Scenario 1 (Mobile content delivery) As referring to Figure 1, suppose two users, A and B, have MHDs and now stay in the same network domain, for instance, WLAN. A has some media objects in her MHD and wants to share them to B. A's MHD first discovers B's MHD in its surrounding, and then negotiates with B's MHD to start the media object transfer. In case that during transferring, B must leave to another place, but the transfer has not finished. At this moment, A decides to redirect B's MHD to download the rest from her home server. A's MHD immediately communicates with the home server, via either 3G or WLAN/wired network, to ask for preparing a secure download transaction. A then provides B specific certificates that are required to process secure transaction with her home server. After leaving the WLAN, B's MHD can download the rest from A's home server through other available networks, notwithstanding A and B are not in different network domains.

Accordingly, we take into account several design requirements and propose a mobile content delivery mechanism in such an integrated, hybrid network environment. The mechanism design comprises three basic components: *device discovery*, *mobile content delivery*, and *secure double-key access control*.

Specifically, as a precedent phase to mobile content delivery, it is prerequisite to employ a device discovery technique to find neighbor networked devices within a network domain. Our study examines several discovery protocols in the literature [1][2], as will be mentioned in Section 3. Among others, the Universal Plug and Play (UPnP) device architecture is used to make clusters of networked devices interconnected in ad hoc or unmanaged networks, especially in domestic environments [3]. Note that our companioned work has fulfilled an AV media content sharing framework based on the UPnP middleware in a home computing network environment [4]. It can basically provide the device discovery as considered herein. Our work further aims to relieve the location restriction, apart from local area network to anywhere, and integrates substantials to achieve the mobile content delivery in integrated network environments.

Ordinary media content transfer between MHDs needs complicated manual setup steps and various data cables or connectors, empirically, the event which is difficult and unfriendly for end users. At present, although many MHDs have Bluetooth functions enabling wireless data transfer without manual wiring, there is a restriction for MHDs of which media content provider and receiver must stay closely in the same Bluetooth PAN. Particularly, point-to-point transfer between two MHDs should be in the same administrative network. The movement of MHDs is confined undesirably. To mitigate it, our design is to use the role of home server on behalf of mobile content provider. The collaboration between the mobile content provider and its home server supports the *mobile content deliver*. Explicitly, in our design, the mobile content provider can have alternative to conduct the receiving client to access indicated media contents from its home server where the mobile content provider reposits duplicates. As a result, this mechanism can have thrice benefits. First, a home server in the wired network can provide higher data throughput and shorten the transmission duration. Second, the mobile content provider can avoid long transmission duration and so reduce energy expense. Finally, both the mobile content provider and the receiving client can get rid of the movement confine from a single-hop transmission range. They are free to move in the network. The stationary home server is reachable by the receiving client. So, it can continue to access media contents from the home server, even though the mobile content provider has roamed to another network or detached the network.

Another effort is to develop a *secure double-key access control* technique among the mobile content provider, home server and mobile content receiver. It is essential for the home server to be able to determine whether a mobile content receiver is trusty or not. We design a simple *double key identification* with two symmetric keys, i.e., provider key and transaction key. The provider key is pre-determined and kept by the mobile content provider. It maintains the trustiness between a mobile content provider and its home server. In contrast, the transaction key is transient and assigned a valid deadline. In response to every content delivery, the mobile content provider applies to its home server for a transaction key to each transaction. This key is used to secure the temporary transaction between the mobile content receiver and home server. Note that this transaction key is opaque to the receiver and will be invalidated immediately after the transaction is accomplished. This enforcement prevents distributing a transaction key to unauthorized receivers. In addition, the provider key may be refreshed to avoid being cracked whenever a mobile content provider finished a transaction.

This paper describes the design and the development reference of a secure mobile content delivery mechanism, which integrates device discovery, mobile content delivery and secure access control software components, in an integrated, hybrid network environment. The UPnP middleware is employed to function as the networked device and service discovery. The role of a home server is introduced to fulfill mobile content delivery on behalf of the mobile content provider. A light-weight secure access control technique with simple double key identification is devised to guarantee the trusty transaction process. Further, our prototype implementation successfully demonstrates the feasibility and effects of the proposed mechanism.

The rest of this article is organized as follows. Section 2 mentions the design overview of the proposed mechanism, and its components and functions. Section 3 briefly describes the background knowledge of discovery protocols, and the UPnP technique. Section 4 describes the mobile content delivery mechanism, and the associated secure access control technique is mentioned in Section 5. Section 6 shows the prototype development. Conclusions are given in Section 7.

2. Mechanism Design Overview

The design objective is to make networked devices interconnected and to enable mobile content delivery anywhere in a secure and trusty fashion within an integrated, hybrid network playground. Consider the proposed user scenario: a user can use an MHD to easily and conveniently deliver stored media contents to other networked devices, or to alternatively redirect them to download indicated media contents from the home server. To bear up this scenario, all mobile content *provider*, *receiver* and *home server* involved are required to support the proposed mechanism.

As mentioned before, this mechanism consists of three basic functional software components: UPnP-based discovery middleware, mobile content delivery service, and secure access control technique. The UPnP middleware functions the device discovery service in an LAN, WLAN, ad hoc, or singly administrative network. UPnP-compliable devices can discover and learn what services each device supports in the network. In addition, the UPnP middleware provides the IP networking and HTTP-based communication protocols. The application-level mobile content delivery service provides file access procedures, such as browse or download actions, as well as necessary information delivery. It performs in a remote procedure call (RPC) methodology that is commonly used in wide, distributed network systems [5]. In addition, a secure access control scheme with double key identification employs the notion of secure transaction process to make the guarantee of access authentication and authority among the home server, mobile content provider and receiver. With provider key and transaction key, respective signatures applied to any transaction-based access process simply identify their owners and keep the trustiness in a short transaction duration.

3. UPnP-Based Discovery Framework

This section surveys several discovery protocols in the literature. The UPnP technique is mentioned especially with its potential problems.

3.1. UPnP Networking Framework

Device and service discovery is essential to enable devices and services to properly discover, configure, and communicate with each other. Specifically, discovery is a mechanism for dynamically referencing a resource on the network. Clients find resources automatically rather than needing pre-configured bindings to specific resources. In practice, it is designed to minimize administrative overhead and increase usability. The literature [1][2] reviews several discovery protocols. We accordingly derive that the UPnP technique is better designed for deployment in an ad hoc or a singly administratively network context. The UPnP technology provides a distributed, open networking architecture that leverages TCP/IP, HTTP and Web technologies to enable seamless proximity networking, control and data transfer in ad-hoc or unmanaged networks. It defines two device categories, control points (CP), and controlled devices or simply "devices." A controlled device functions as a server, offering services that can be monitored or controlled by a CP. The UPnP messages are hosted in UPnP-specific Internet protocols such as Simple Service Discovery Protocol (SSDP) [6], General Event Notification Architecture (GENA) [7], Simple Object Access Protocol (SOAP) [8], and are transferred by HTTP over TCP or HTTP unicast/multicast over UDP. Ultimately all messages are delivered over IP.

A UPnP device performs six function layers, specifically, as follows in a bottom-up order.

- *Addressing* is an underlying function enabling a device to acquire a unique network IP address when it newly joins a network. The default addressing method is dynamic host configuration protocol (DHCP), and automatic IP configuration [9] is used as the absence of DHCP service.
- *Description* is used for a device to summarize its services and capabilities in a well-defined format, so control points can parse description files and know what a device offers.
- *Discovery* is the capability for a device to advertise its appearance and services in the network, so a control point can find the device and its information.
- *Control* is the capability for a device to handle requests from control points and to invoke specific actions.
- *Eventing* is used by a device to notify the registered control points whenever any interested state changes.
- *Presentation* is an HTML-based interface provided by device for users to control or monitor them directly.

3.2. UPnP Device and Service Discovery

Our design merely leverages the network addressing, discovery, description and control function layers in partial, but excludes the eventing and presentation layers. Note that in our design, the UPnP functionality on a mobile content provider is symmetric to that on a mobile content receiver.

Figure 2 illustrates the UPnP discovery procedure. As UPnP devices power on, they first check the existence of DHCP server in the network domain. If not, they use automatic IP assignment instead to configure dynamic assignment of IPv4 link-local addresses in 169.254/16 range. After network hosts have IP address in the same network domain, they proceed to discover each other.

The UPnP discovery mechanism is based on the SSDP which is a simple HTTP-based discovery mechanism that discovers local resources in a small, local area network with no need of centralized configuration, management, or administration. A UPnP device periodically advertises its the appearance on a well-known address/port, 239.255.255.250:1900 (SSDP:NOTIFY), i.e., a HTTP multicast over UDP. Every active device in the network can be aware of its appearance. In addition, each device can directly query the network (SSDP: SEARCH), and each resource host can directly respond to the request (HTTP/OK Response). Note that the HTTP LOCATION headers in advertisement and response messages specify the URLs to the same description of the UPnP root device. A UPnP device uses a description to present its services and capabilities in a well-defined XML format. So, interested UPnP devices can fetch and parse the description, and know what services the device offers as well as its profile information.

The UPnP control is customized according to the SOAP. The SOAP integrates both HTTP and XML to provide a Web-based messaging and remote controlling mechanism. Our design preserves this function layer by thinking that the UPnP Forum has standardized several device control profiles [10] to make consensus on the activities of different device categories. For example, UPnP AV profile [11] is applied to instantiate any CE device that streams AV content to other UPnP devices. In support of these profiles, the MHDs can extend their application scope to the dimension of home automation and entertainment. It is observed however that mobile content delivery service is not included yet. Rather, this mechanism adopts a "customized control profile" that can satisfy our requirements about content browse and meta-data retrieval.

Incidentally, eventing and control layers are investigated to be not beneficial to the development of mobile content delivery in hybrid network environments. Eventing based on the GENA performs in a publisher-subscriber method, which can be useless since MHDs are not set with static network addresses empirically in wireless and mobile network systems. As for the control layer, it functions by means SOAP, an application-level communication protocol over HTTP. The SOAP messaging rests on reliable network connections that should be attainable in WLAN or small area networks. However, it is heavy and can induce higher computation overhead. In fact, it is advisable for Web services, but not for general ordinary remote network services, especially in unreliable wireless and mobile network environments as considered.

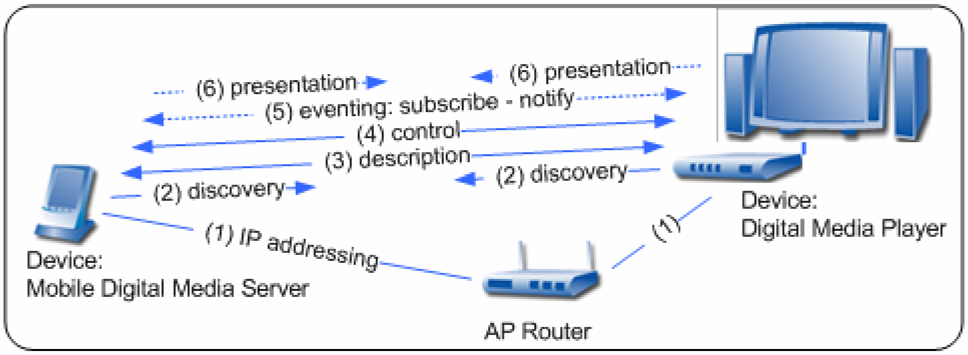

Figure 2. The interaction procedure in a UPnP network environment.

4. Mobile Content Delivery

This mechanism proposes three basic methods of mobile content delivery service; thereof they provide complementary usages. In addition, a *content directory service* is designed to render the content presentation friendly to mobile content receivers.

4.1. Content directory service

The content directory service checks all shared media items and summarizes their information into meta-data. Our work customizes this service to meet our requirements of media content management. Consider the native file system, provided by the mobile handheld platform, is usually too simple to satisfy developers' and users' demands. We develop this service with friendly and flexible UI structures for end users to process media content browsing and other operations. For instance, it can catalog media items into image, audio, video or document directories according to their formats rather than mixing them into a plain directory. To ease exposition, the below presents the meta-data structure of every media object in our design.

```
  <ItemList>
     <Item  Type="#{TypeName}" /* directory or file */
               Name="#{FileName}"
               Size="#{SizeInBytes}"
               URL="#{DownloadURLReference} "/>
 </ItemList>
```

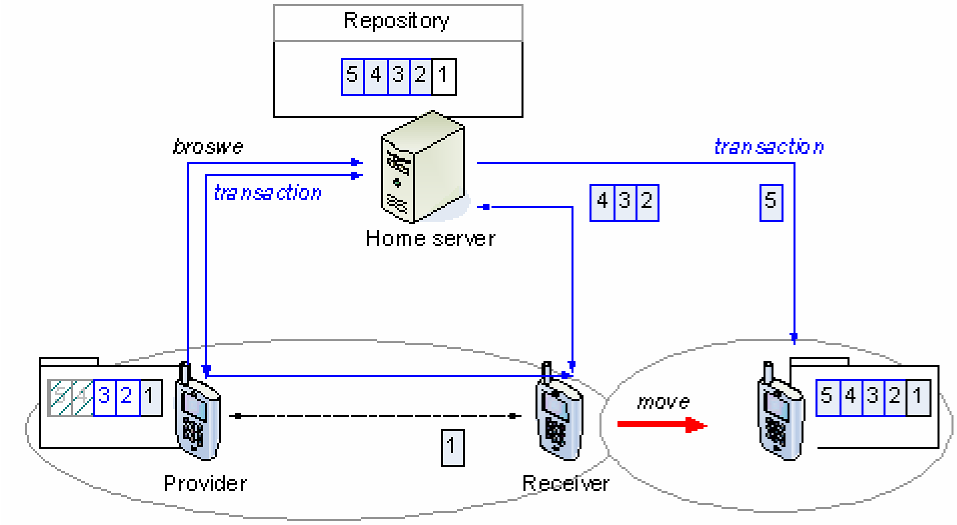

Figure 3. The concept model of mobile content delivery.

4.2. Direct downloading (Case 1)

The mobile content provider and receiver process the basic UPnP logic to discover each other. The receiver obtains both device and service description files of the mobile content provider. With the customized control profile, the receiver is able to *browse* the content directory of the mobile content provider, and then obtains a list of "shared" media objects the mobile content provider intends to offer. The mobile content receiver accordingly acts an HTTP GET to the reference URL from which an indicated media object is downloaded. By the way, the receiver can download the content in an RPC alternative, later described in Cases 2 and 3, if it is supported by both of the mobile content provider and receiver.

4.3. Redirect downloading (Case 2)

This mechanism designs a redirect downloading method used to enhance the system performance, in terms of utilization of battery energy and network bandwidth. A mobile content provider likely instructs the receiver to access media content from its home server instead, while the transfer of all indicated media objects has not been downloaded completely. Particularly, as in Figure 3, the provider owns five media objects, 1, 2, 3, 4 and 5, in its home server, but partially keeps objects, 1, 2 and 3, in its local storage due to limited storage, object deletion or other reasons. The provider queries its home server about a meta-data list that contains a number of location references to media objects stored there. By comparing the local list to the receiving list, the provider forwards the receiver a "working list" of indicated media objects it redirects the receiver to download from the home server. Note that the working list can contain some or all of local items if the home server has their copies. As the illustrative example in Figure 3, only object 1 is delivered by the provider, though objects 2 and 3 are locally available. The receiver can download objects 2, 3, 4 and 5 from the home server through another network, different from the one used to interconnect the provider.

4.4. Continual downloading (Case 3)

Our design provides a continual downloading method in support of mobile content delivery after mobility or connection reestablishment by either the mobile content provider or receiver, provided that transaction deadline has not passed, or that "downloading intermission" is not longer that a specific interval. While the mobile content receiver keeps the working list that was assigned by the preceding redirect downloading process, it can still make connections to download indicated media contents till those location references are invalidated by the home server. Herein, note that redirect downloading and continual downloading procedures are subject to a secure transaction scheme, as will be jointly detailed in the next section.

5. Double-Key Secure Access Control

The double-key secure access control scheme is designed based on the notion of transaction process among the mobile content provider, receiver and home server. Figure 4 shows the conceptual procedure. The mutual relationship of trustiness is guaranteed by means of double key identification with a pair of provider key and transaction key.

5.1 Transaction Process

This mechanism imposes a secure transaction process with authorization and authentication among the mobile content provider, receiver and home server. Definitely, when the provider commences a redirect downloading process, a secure transaction process is initialized. The home server manages the transaction process since it is the content provider instead. A transaction process is assigned with a transaction or *session* deadline. The "session" is used to mean the receiver can asynchronously perform plural continual downloadings to complete the working list before the transaction deadline. Moreover, any downloading intermission, before the session deadline expires, must not be longer than a specific serving interval.

Consequently, a secure transaction is effective from the moment at which the provider inform both its home server and the receiver till the end of the last media item is downloaded completely, except that any of the following situations occurs hereinto: any participant aborts the transaction process; the associated session deadline expires; the intermission between two successive continual downloadings exceeds the serving interval.

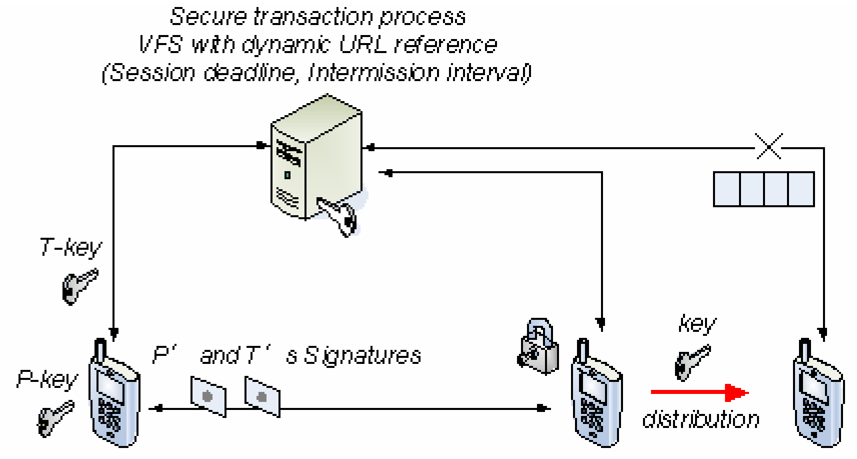
Secure transaction process
VFS with dynamic URL reference
(Session deadline, Intermission interval)
T-key
P-key
P' and T' s Signatures
key
distribution

Figure 4. Double-key secure access control.

5.2. Double Key Identification

In order to ascertain whether a mobile content receiver is trusty or not, our mechanism design proposes a simple double key identification scheme with a pair of symmetric keys, i.e., provider key and transaction key.

- *Provider key* (P-Key) is pre-determined and kept secure by the mobile content provider. It maintains the trustiness between a mobile content provider and its home server. The provider key may be refreshed to avoid being cracked after a mobile content provider finished a transaction.
- *Transaction key* (T-Key) is dynamically generated by the home server whenever a new transaction is initialized. This key is used in the transaction between the mobile content receiver and home server. It is merely effective to the end of the session deadline. Specifically, in regard to every redirected downloading, the mobile content provider asks its home server a transaction key which is opaque to the mobile content receiver and will be invalidated immediately after the transaction is accomplished. This enforcement prevents distributing a transaction key to unauthorized receivers.

The double key identification scheme and its usages in a secure transaction process are explained below. To facilitate comprehension, readers may please refer to Figure 5, the flowchart of interaction procedure in the later section.

This mechanism adopts an RPC-like model to conduct the transaction process between the home server, and mobile content provider or receiver. Two field entities, *identifier* and *signature*, are specified in order to facilitate the resolution of authentication and authority between RPC sender and receiver. They are associated with the URL reference, indicated by an RPC invocation. Precisely, an RPC invocation expression should contain the sender's identifier, by which the receiver determines if the sender is authenticated and trusty, or if the sender is authorized to execute indicated procedure. In addition, this expression may include the sender's signature to convince the receiver that some critical data enclosed is intact. The receiver can compare the received signature with the exact one to make the decision. The canonical forms of identifier and signature entities in the URL expression are shown below.

```
Identifier := #{P-Key} | ( #{P-Key} & #{T-Key} )
Signature := MD5( #{DigestURL_Raw} )
```

The identifier is not exposed under the expression. It is added in the DigestURL_Raw expression that is further hashed by the MD5 function to generate a signature. Both P-Key and T-Key values are unique in the meanwhile of a transaction process, and so is the resultant signature. Accordingly, the receiver can believes the sender to be trustworthy.

The DigestURL_Raw element has three different compositions, DigestURL_Raw_0, _1 and _2, as the inputs of MD5 function for signature generation in support of different sorts of RPC invocations. The first two are enclosed in the RPC invocation expression sent from the provider to the home server. The last is enclosed in the RPC invocation expression sent from the receiver, by the provider, to the home server. The provider takes charge of subscribing the corresponding MD5(DigestURL_Raw_(0|1|2)) at the end of the URL expression.

```
URL := "HostURLBase/RPCAction?
        File=#{FileName}&
        User=#{UserName}&
        Signature=#{Signature}"
```

```
DigestURL_Raw_0 := "/RPCAction?
                    User=#{ProviderName}&
                    Key=#{P-Key}"
DigestURL_Raw_1 := "/RPCAction?
                    File=#{FileName}&
                    User=#{ReceiverName}&
                    Key=#{P-Key}"
DigestURL_Raw_2 := "/RPCAction?
                    File=#{FileName}&
                    User=#{ReceiverName}&
                    Key1=#{P-Key}&
                    Key2=#{T-Key}"
```

In comparison with URL and DigestURL_Raw expressions, notice that the URL eventually refers to the host that runs the indicated RPCAction routine upon the target file objects. The HostURLBase element designates the host's location reference, of home server or mobile content provider, depending on which downloading method is used. Particularly, in the case of redirect or continual downloading, the HostURLBase value records the home server. Otherwise, it refers to the mobile content provider itself. By the way, the UPnP device description may contain an optional HostURLBase element that reveals the host location reference. Furthermore, according to which RPCAction to be invoked, the mobile content provider must subscribe an appropriate signature, by one of DigestURL_Raw_0, _1 and _2. There is an initial set of RPC actions specified for the secure mobile content delivery, as will be mentioned in the next section. In addition, there are several statements below. The major difference between DigestURL_Raw_0 and DigestURL_Raw_1 is the value of User field that means where an RPC action originates from. The value of Key=#{P-Key} in DigestURL_Raw_1 means that this RPC action is committed by the provider. The pair of P-Key and T-Key in DigestURL_Raw_2 means both the provider and the home server commit themselves to this RPC action for the receiver. For instance, continual downloading is such a case: the provider asks the home server in advance to permit the receiver’s downloading request. Yet, the home server can deny any RPC actions.

5.3. More Remarks about Double-Key Identification Scheme

Using the double-key identification scheme implies that our mechanism utilizes the signature technique to verify the RPCAction requests instead encrypting those requests themselves. In practice, the use of signature is enough to protect the originality and integrity of the RPCAction invocation requests. Particularly, remind that the home server knows the provider and the receiver names, supports a list of RPC actions and keeps P-Key and T- Key secretly. It is so able to generate a signature and compares this with the received one. If two signatures are not equivalent, the home server rejects the RPCAction invocation. Incidentally, basic MD5 hash function is adopted in the present prototype, but the choice of other robust hashing candidates may be important as a considerable future work.

Therefore, this scheme design does not adopt sophisticated algorithms to generate a symmetric key or shared secret key interchangeably. Rather, it uses symmetric-key algorithms which are generally much less computationally intensive than asymmetric key algorithms. In spite of that one disadvantage of symmetric-key management is the requirement of a shared secret key, with one copy at each end, the impact of a potential discovery by a cryptographic adversary would be negligent because only three participants involved in a transaction process. Moreover, a transaction key is transient, and different transactions are assigned with different keys. In addition, the provider key can be changed regularly with the home server and kept secure during distribution and in service. Furthermore, if necessary, asymmetric algorithms are used to distribute symmetric-keys at the start of a session. This simplifies the key distribution problem, because asymmetric keys only have to be distributed authentically, whereas symmetric keys need to be distributed in an authentic and confidential manner.

6. Prototype Demonstration

This section describes the experimental prototype implementation. To ease exposition, two real cases of direct downloading and redirect downloading are developed respectively to demonstrate the interaction procedure in the mobile content delivery.

The prototype framework comprises two major software components: the UPnP stack and RPC-like Web service modules. The UPnP stack module has been implemented in [4]. All RPC-like interaction processes between the mobile content provider and its home server is developed by using application-level Web communication technologies. Interactive messages are represented in XML conventions. In addition, the mobile content receiver requires a tiny customization or software installed to be able to interact with the home server over the HTTP communication protocols. Therefore, any networked devices having TCP/IP connectivity, Web server or client, and XML functions can perform the mobile content delivery service in the proposed framework.

Specifically, the experimental prototype runs in a RPC logic. Considering the interactive behavior, this mechanism suggests an initial set of RPC operational primitives for messaging interactions among mobile content provider, receiver and home server, and for establishment of a secure mobile content delivery. Table 1 shows the set of RPC actions that have a uniform usage as URL := "HostURLBase/RPCAction?Parameters. To better grasp the usages of RPC actions in Table 1, two interactive procedures in direct and redirect downloading cases are described below, as shown in Figure 5.

In an initial phase, both mobile content provider and receiver use the UPnP discovery function to find each other and negotiate in preparation for mobile content delivery. In reference to Section 3 and [4], this UPnP phase is not exhibited. After the provider and receiver are engaged, the phase is to browse the content directory and download indicated media objects. In case of direct downloading, the receiver invokes a Browse action on the provider and receives the response of a working list of meta-data records, arranged by the content directory service with actual URLs. The receiver applies HTTP GET iteratively to download separate media objects from the provider.

```
Browse:
    <ItemList>
        <Item Type="File" Name="Track1.mp3" Size="3155202"
            URL="/140.115.152.2/AV_Dir/Music/Track1.mp3"/>
    </ItemList>
Download File:
    HTTP GET /AV_Dir/Music/Track1.mp3 HTTP/1.1
    HOST: 140.115.152.2:50988
```

In case of redirect downloading, the provider invoke a Browse action on the home server and receives the response of a directory list of meta-data records. Then, the provider does Redirect action to notify the receiver of preparing the following redirect downloading *transaction*, and then forwards it a newly working list of indicated meta-data records. Every item in the new list is same as the above, note worthily except the value of the URL element that distinguishes the home server from the provider.

The receiver is obligated to do ApplyForDownload or ApplyForBatchDownload action in advance of retrieving any files from the home server. The receiver informs the provider a list of indicated media objects it wants to access from the home server. The provider applies its signature MD5(DigestURL_Raw_1) onto every URL and in turn a ApplyForPermission request to the home server. If the provider's signature is confirmed, the home server generates a transaction key and initializes a secure *transaction* process to deal with this request. Subsequently, the transaction key and a list of temporary URLs are replied to the provider. The provider further modified every temporary URLs by apply a joint signature MD5(DigestURL_Raw_2) onto the URL. Then, it sends the receiver a final working list as a reply to ApplyForDownload action. Accordingly, the receiver is able to perform DownloadFile actions to asynchronously retrieve indicated media objects

Table 1. An initial set of RPC actions.

RPCAction Name	Description and Usage
Browse	browse the content directory service on the hostURLBase site /Browse?User=#{(ProviderjReceiver)}
Search	search any specific media object on the indicated site /Search?User=#{(ProviderjReceiver)}

Redirect	redirect the receiver to change the downloading method
	/Redirect?User=#{Provider}
	&Signature=#{MD5(DigestURL_Raw_0)}

ApplyForPermission	apply to home server for downloading permission
	/ApplyForPermission?File=#{FileName}
	&User=#{Receiver}
	&Signature=#{MD5(DigestURL_Raw_1)}

ApplyForDownload	apply to provider for redirectly downloading a file
	/ApplyForDownload?File=#{FileName}
	&User=#{Receiver}

ApplyForBatchDownload	apply to provider for redirectly downloading files
	ApplyForDownload?File=#{(FileName)+}
	&User=#{Receiver}

DownloadFile	download _les via HTTP GET in redirect or continual mode
	/Download?File=#{FileName}
	&User=#{Receiver}
	&Signature=#{MD5(DigestURL_Raw_2)}

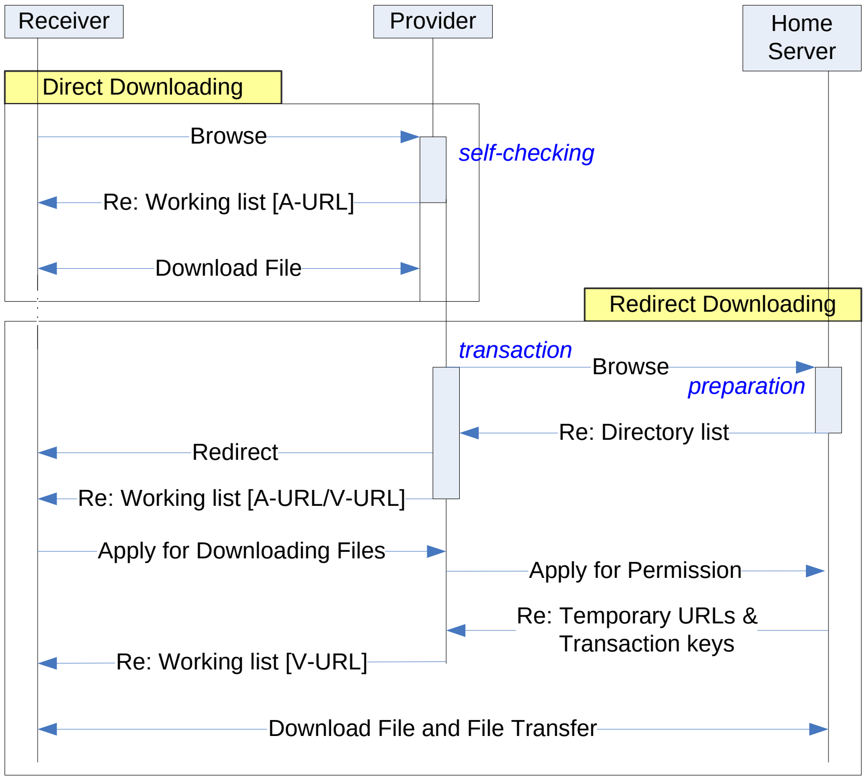

Figure 5. Interactive flowchart.

7. Conclusion and Future Work

The work in this paper has devised a secure mobile content delivery mechanism which is able to make networked devices interconnected and to enable mobile content delivery anywhere in an integrated network playground. To this purpose, we have described the design and development of several significant components: UPnP discovery middleware, mobile content delivery service, and transaction-based double-key access control scheme. On top of this framework, we have defined an initial set of RPC-based operational primitives for establishing mobile content delivery services as scenarios addressed in this paper. Furthermore, an RPC-based experimental prototype is successfully demonstrated.

Currently, we are investigating several design issues, development challenges and requirements on the proposed mechanism. The future work will include three aspects. First, we will extend the set of RPC-based actions to support publish/subscribe services with event notification. The intention is to develop a novel mobile content synchronization service. Second, we will design service management, storage I/O, manipulation and local environment primitives. They will be utilized to provide the basic requirements for developing a mobile device control paradigm. Finally, additional security packages and technologies will be examined and applied in place to strengthen the system security and robustness.

8. Acknowledges

The work was supported in part by the National Science Council of Taiwan, R.O.C., under Contracts NSC97-2218-E-008-009.

9. References

[1] W. K. Edwards, "Discovery Systems in Ubiquitous Computing", *IEEE Pervasive Computing*, April-June 2006, pp. 70–77.

[2] F. Zhu, M. W. Mutka and L. M. Ni, "Service discovery in pervasive computing environments", *IEEE Pervasive Computing*, Vol. 4(4), October-December 2005, pp. 81-90.

[3] UPnP Forum, "UPnP Device Architecture 1.0 Version 1.0.1", December 2003.

[4] Chih-Lin Hu, Wen-Shun Liao and Yen-Ju Huang, "Mobile Media Content Sharing in UPnP-Based Home Network Environment", *Journal of Information Science and Engineering*, Vol. 24, No. 6, November 2008, pp. 1753-1769.

[5] D. Comer and D. Stevens, "Internetworking with TCP/IP Vol. III (Linux/POSIX Socket Version)", 2000.

[6] IETF Internet-Draft, "Simple Service Discovery Protocol/1.0 Operating without an Arbiter", IETF Internet-Draft draft-cai-ssdpv1-03.txt, October 1999.

[7] IETF Internet-Draft, "General Event Notification Architecture Base: Client to Arbiter", IETF Internet-Draft draft-cohen-gena-pbase-01.txt, September 2000.

[8] W3C Consortium, "Simple Object Access Protocol", W3C Consortium: www.w3.org/TR.2000/NOTE-SOPA-20000508, May 2000.

[9] IETF Internet-Draft, "Dynamic Configuration of IPv4 Link-Local Addresses", IETF draft-ietf-zeroconf-ipv4-linklocal-17.txt, July 2004.

[10] UPnP Forum, "Standardization Device Control Protocols", http://www.upnp.org/standardizeddcps/.

[11] UPnP Forum, "UPnP AV Contentdirectory:2 Service Template Version 1.01", May , 2006.

[12] Y. Liong and Y. Ye, "Effect of UPnP Advertisements on User Experience and Power Consumption", In *Proceedings of the 2nd IEEE Consumer Communications and Networking Conference (CCNC'05)*, Janunary. 2005, pp. 91– 97.

[13] K. Mills and C. Dabrowski, "Adaptive Jitter Control for UPnP M-Search", In *Proceedings of the 38th annual IEEE International Conference on Communications (ICC 2003)*, Volume 2, May 2003, pp. 1008–1013.

[14] R. Fielding, "Architectural Styles and the Design of Network-Based Software Architecture," on-line available via URL: http://www.ics.uci.edu/fielding/pubs/dissertation/top.htm.

[15] J. Newmarch, "A RESTful Approach: Clean UPnP without SOAP", In *Proceedings of the 2nd IEEE Consumer Communications and Networking Conference (CCNC'05)*, January 2005, pp. 134–138.

[16] G. Bieber and J. Carpenter, "Introduction to Service-Oriented Programming," on-line available via URL: http://www.openwings.org/download.html, Sept. 2001.

[17] Y. Mazuryk and J. J. Lukkien, "Analysis and Improvements of the Eventing Protocol for Universal Plug and Play," in *Proceedings of the IASTED Conference on Communications, Internet and Information Technology*, Nov. 2004.

[18] C.-L. Hu, Y.-J. Huang, and W.-S. Liao, "Multicast Complement for Efficient UPnP Eventing in Home Computing Network.", In *Proceedings of IEEE International Conference on Portable Information Devices (Portable'07)*, Feb. 2007.

Authors

Chih-Lin Hu received the BS degree in computer science from the National Cheng-Chi University in 1997, the MS degree in computer science from the National Chung-Hsing University in 1999, and the PhD degree in electrical engineering from the National Taiwan University in 2003. He was a researcher at BenQ and Qisda Advanced Technology Centers, Taipei, from 2003 to 2007. Since 2008, he has been an assistant professor in the Department of Communication Engineering, National Central University, Taoyuan, Taiwan, R.O.C. His research interests include broadcast information systems, mobile data management, home networking technology, mobile agent technology, and mobile and pervasive computing systems. He had the honor to get the best paper award in IEEE ICPADS 2000 and BenQ Innovation Awards in 2006 and 2007. Dr. Hu is a member of the IEEE.

Chien-An Cho is an undergraduate student in the Department of Communication Engineering, National Central University, Taoyuan, Taiwan, R.O.C. He obtained the award of NCU outstanding students scholarship in 2008. His study interests range widely from mobile communication technology, computer networking protocols, Internet and Web applications, and GUI design patterns for end users on various target devices. Mr. Cho is an IEEE student member.