Using Secure Device Discovery and Device Authentication in Unsecure Gaming Network

Janne Väre, Vesa Pehkonen, and Katja Pulkkinen

***Abstract*—A remote gaming application Games@Large needed to have an automatic device discovery in an environment where network is not trusted and neither are client devices used in the network. We applied earlier invention Secure UPnP (Universal Plug and Play) to Games@Large demonstration platform to enable secure device discovery, device authentication and communication in an untrusted network. The Secure UPnP method was improved by adding possibility to work in an unsecure network while earlier only problem of untrusted devices was addressed. As a result of the addition of the Secure UPnP to the Games@Large gaming platform it can now be used in unsecure environments like internet café. The Secure UPnP method still needs development as for example certificate expirations are not handled and it could still be more user friendly to customer users and administrators.**

***Index Terms*—Device discovery, remote gaming, security, UPnP.**

I. Introduction

This paper describes a method where an earlier invention, Secure UPnP is slightly modified and applied in a Games@Large (GaL) project demonstration. Secure UPnP was developed to have a method to allow access only to certain devices to a network and still to enable UPnP discovery and allow a device to communicate with other trusted devices. A problem solved by the method was to combine easy access with little configuration needs with high security. The improvement made to the Secure UPnP method for this demonstration was that in addition to handling untrusted devices also the network may be unreliable. Games@Large, a remote gaming project demonstration utilizes UPnP device discovery and it has running environments like internet café where the network is considered unreliable and all client devices connecting to the network cannot be trusted either. The use of Secure UPnP moderates what client devices can access the Games@Large system and also makes sure that authorized clients can find only the real Games@Large server(s).

The same method can be used in all other platforms and environments where automatic service and device discovery is required and where network and client devices are considered unreliable like is the case usually in open wireless networks. The method can also be applied to other discovery protocols than UPnP.

This paper consists of description of Games@Large project, Secure UPnP method and how it is used in the Games@Large demonstration.

Manuscript received April 15, 2012; revised May 1, 2012. This work has been carried out in the IST Games@Large project [1], which is an integrated Project under contract no IST038453 and is partially funded by the European Commission.

Authors are with VTT Technical Research Centre of Finland, P.O. Box 1000, 02044 Espoo, Finland (e-mail: firstname.lastname@ vtt.fi).

II. Games@Large Project

Games@Large (GaL) is a European Information Society Technologies (IST) Integrated Project intending to research, develop, and demonstrate a platform for remote gaming. The GaL platform intends to make digital games ubiquitously available to users in different environments such as homes, hotels, elderly houses, cruise ships, trains and internet cafés. Unlike conventional gaming systems, the GaL platform will not be restricted to be used by dedicated devices from a single manufacturer but to provide support for different kinds of standard devices. In addition to PCs and laptops, these include devices with limited technical capabilities such as set-top-boxes (STB), enhanced media extenders (EME) and handhelds. The GaL architecture will be based on distributing the game processing between the game server and the client devices (Fig.1). The game server which is a normal PC (or several of them) runs the game software and uses network connectivity (wired or wireless) to deliver the game to the user. The game contents are streamed through the network using either pre-rendering or video streaming technologies developed in the project. Software in the client device decodes the received stream, presents the contents on the screen and forwards the user's game commands to the server. [2]

Fig. 1. Games@Large enterprise edition framework [1]

A user that interacts with the Games@Large system can use his/her own device or client device provided by the system (e.g. STB) to browse, select and play games. Besides of the main functionality GaL system has a lot of background activities such as discover the devices in the network, authenticate and authorize the device and the user, log the activity in the system for tracking and billing, configure and maintain the system and so on.

The server side of Games@Large consists of Local Management Server (LMS), Local Processing Server (LPS) and Local Storage Server (LSS). There can be one or more LPS PCs or it can be integrated in LMS PC. LSS is a virtual component and usually means disk storage of LMS.

III. Games@Large Device Discovery

One of the main design criteria of Games@Large was simplicity as well in design as in usage of the system. The user should not need to do any configuration or searches before starting to use the system. As the user may use different devices to play an automatic device (and service) discovery was required.

Several different device discovery techniques existed, but for Games@Large usage of UPnP device discovery was proposed to ensure compatibility with several commercial products, and follow a common accepted standard. Using UPnP also allows the network and its elements to be used for other entertainment purposes, like watching videos and listening to audio.

The Device Discovery process in high level consists of following actions:

1) Device receives its IP address using DHCP (Dynamic Host Configuration Protocol).
2) Device advertises itself using UPnP multicast advertise messages
3) Control point (e.g. management terminal) can also search devices and services with UPnP multicast search query messages
4) Control point retrieves device/service descriptions
5) Presentation page using web browser is viewed to user. [1]

IV. UPnP

Universal Plug and Play (UPnP), defined by UPnP Forum and used by DLNA (Digital Living Network Alliance), is a set of protocols that compose technology for flexible and standardized connectivity between networked electronic devices.

UPnP makes it possible that devices can automatically join the network of devices, and announce their presence and service capabilities to various control points in the network. A new device that enters the network can be a physical appliance or just software that is being loaded into a physical appliance that is already in the network. A typical application of UPnP is a local home network where appliances such as a TV, a mobile phone, a media player, and a PC interact using the UPnP architecture. [4] For more information how it is used by DLNA see [5].

V. Secure UPnP

UPnP does not generally provide any security and assumes that only trusted devices have access to the network. For networks where untrusted devices are allowed and only devices controlled by trusted persons should have access to some services, a secure UPnP network architecture, including key management has been developed. The Secure UPnP network architecture is shown in Fig. 2. The architecture uses Transport Layer Security (TLS) to secure all TCP traffic, which carries most of UPnP messages. To establish a TLS session, each node must have an X.509 certificate for authentication. Certificates are granted by a local Certificate Authority (CA) but only if the Administrator has accepted the new node. UPnP discovery phase uses UDP where it is not possible to use TLS, but UDP data is (optionally) encrypted using standard symmetric encryption algorithms. The UDP encryption key is shared by the whole network and is stored in a key server from where it is distributed using TLS. Secure UPnP method adds TLS layer and UDP encryption layers to the standard UPnP protocol stack [3]. OpenSSL library functions are used in all security algorithms. For more information about the certificate handling and encryption algorithms see [6].

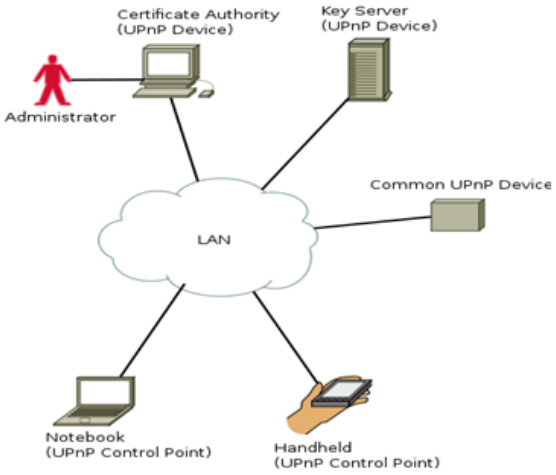

Fig. 2. Secure UPnP network architecture. [3]

VI. Components of Secure Discovery in the Demonstration

Following chapters describe the components needed by the secure discovery of the Games@Large demonstration. Certificate Authority (Server) and Key Server are familiar components from the earlier Secure UPnP demonstration, but also their functionally has been modified as the Secure UPnP method has evolved. Component interactions are shown in Fig. 3.

A. Certificate Authority Server

Certificate Authority Server (CAS) is a component which creates (or actually only signs) X.509 certificates. When CAS is running it can be found without Secure UPnP by clients. It will wait certificate requests which have been encrypted. When a request is received CAS reads all valid keys from

Key Storage, tries to decrypt the request with all keys until it succeeds. Certificate is then signed and returned to the requesting client. The key which turned out to be the correct encryption key is deleted from Key Storage.

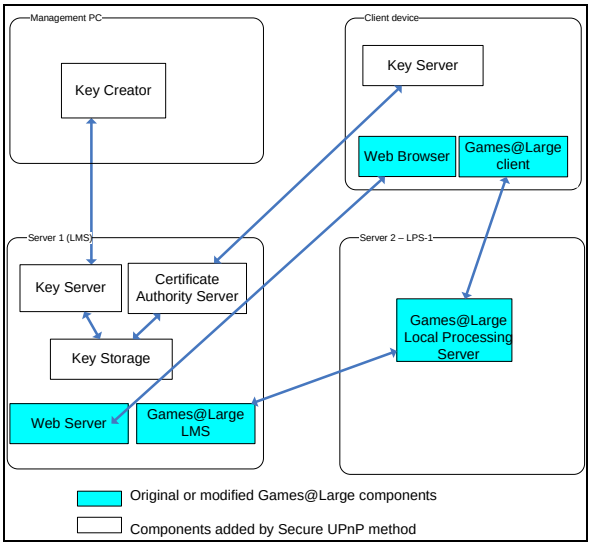

Fig. 3. Components of Games@Large system with Secure UPnP. GaL system consist of one client, one LMS and one LPS (where virtual LSS is integrated).

In current demonstration CAS is used only by the modified Games@Large client software but if secure discovery was extended to be used by server components to discover each other it could be used by them too. The CAS is implemented as console application and it can be compiled to Linux and Windows operating systems. In the demonstration the CAS is installed in LMS PC.

B. Key Storage

Key storage is a component to persistently and securely store keys which are generated by Key Creator and used by other components of this secure discovery system. Each key is accompanied by two timestamps which indicate start and end of validity period.

Key Storage offers following services to other components:

1) Remove Key to remove a key given as parameter from storage
2) Add Keys to insert one or more keys given as parameters to storage
3) Get All Keys to return list of all keys in storage.

The Key Storage has been implemented as a part of complete secure storage solution, which has other purposes like user authorization and authentication. Key storage does not use UPnP or other automatic discovery. It has a GUI from where it can be configured to used TLS or regular socket and chosen what type of database is used. Key Storage is implemented with Qt framework.

C. Key Server

The purpose of Key Server is to store keys. When Key Server is running it can be discovered by Secure UPnP. It will receive one or more keys and stores them to Key Storage address of which has been preconfigured. It is basically a discoverable front end to hide Key Storage which is not discoverable.

D. Key Creator

Key Creator software is used to create keys that are used in encryption of certificate requests and accompanying passwords which are given GaL client users. The key used in this case is 16 bytes length null terminated string. The Key creator allows option to select shorter key resulting in shorter password for easier use in demonstrations. The manager using Key Creator must choose start and end time of period when the keys are valid. After the passwords generated from created keys are ready they are sent to Key Server for storage. They can also be printed on paper to be given to allowed client device users. In the demo version the printed keys are a list of character strings but this could be improved to print nice looking coupons with logos and advertisements.

Because this method is used only for device authentication, the system requires also user authentication e.g. ID and Password. These could be generated at the same time and printed on the same coupon. Additionally if users must connect to a passphrase protected wireless network (which is recommended by GaL security guidelines) this information could also be on the coupon.

VII. Games@Large Components and Their Modifications

A. Web server

The web server – Apache – used by the GaL is modified to require client authentication. This is supported by Apache and it can be done via Apache's configuration file.

B. LPS and LMS

Like mentioned earlier the main server components of GaL are LPS and LMS. They also required a small modification in their UPnP libraries to be able to use the Secure UPnP addition. OpenSSL is already used by GaL for encrypting game control channel and the same library can be utilized by Secure UPnP for security algorithms.

C. Games@Large Client

The GaL client software requires also the same Secure UPnP library modification but also additional code must be inserted to the client executable. The new software part will ask a password from a user with a GUI dialog. The password is used to create an encryption key, which is used to encrypt a certificate request sent to CAS. The created encryption key is of course the same from where Key Creator created the password. In case of an error in getting certificate the software addition has dialogs to inform user about the error situation. GaL client is an UPnP control point.

VIII. Operation of Demonstration

This chapter illustrates operation of the system in general level. All details are not described and it is assumed that there

are no error situations. Normally in case of errors an error message is displayed and the process is stopped.

The GaL system is made Secure UPnP compatible by installing the new components to LMS server, changing UPnP libraries of LMS and LPS, and configuring the GaL Apache server. (Fig.3) Before client users can use the system an administrator utilizes Key Creator software to create keys and from each key also a password is generated. Keys are stored in Key Storage via Key Server and passwords are printed on coupons. Key Storage and Key Server at least must be running during this phase.

Now a new user with a new client device can start using the system. A modified GaL client executable is launched and it then asks the password. User must now input the password as it is printed in the paper coupon and the client then generates encryption key using defined algorithm from the password. If the network uses a protected wireless network user must also input passphrase for it to access the network. The modified GaL client will find CAS service in the network using Secure UPnP and send certificate request which is encrypted by the key. SSL encrypted UPnP control messages are used between the client and the CAS. CAS will verify that the certificate request is valid by trying to decrypt it with every key that is stored in the Key Storage. If a result of a decrypting try is successful the key is valid and a new certificate is created and returned to the modified GaL client. When the client has received the certificate it acknowledges this to the CAS and the key is removed from Key Storage. The client software then stores the certificate in the web browser (Mozilla Firefox) client certificate storage. CAS and Key Storage Processes must be of course running during the certificate creation time.

The modified GaL client then utilizes Secure UPnP to discover Secure UPnP enabled GaL LMS from the network. If this is successful the client receives web name (or IP address if names are not used) of a GaL web server and the client then launches a web browser with URL of GaL web server as parameter. The web name is received in an extension of LMS certificate to be sure that the web server is not fake but real Games@Large server. The URL is generated from pre-configured information (e.g. page name) and the web name.

The client web browser will next connect to the GaL web server. Again both the client web browser and the web server validate each other's certificates before allowing access and displaying the GaL start page. The Games@Large service will run normally now. Game streaming of audio and video or graphics, will be unencrypted and the control channel is encrypted as it was before applying Secure UPnP.

The next time the user uses the GaL system the certificate already exists and the user does not need to input the password anymore. The certificate of course has limited validity period and after this period the user is required to enter a new password for the generation of a new key.

UDP encryption offered by Secure UPnP is not used in this demonstration.

IX. Conclusion

The inclusion of Secure UPnP into Games@Large as described adds one level of security into device discovery and device authentication. It also adds some complexity for user because there is additional input of password. This however needs to be only once for each GaL network used. Otherwise, based on our user tests, the Games@Large system works normally and users will not notice any inconvenience. From managerial point of view there are more changes, the extra software management and key/password generation for all users, but this should not be problem if the added security is required like e.g. in the internet café scenario of Games@Large project. In GaL the use of this method could also have been extended to utilize secure discovery by all components (i.e. LMS and LPS) discovering each other for example when there are several LPS servers in the network but so far this has not seemed necessary as the network segment where servers are can be secured by other means.

The earlier paper [3] discusses problems in Secure UPnP. These are for example lack of certificate revocation lists and methods how certificate and encryption key expirations are handled. These unfortunately are still unsolved but they are solvable. The system is also dependent on reliability of human operators handling passwords and these kinds of threats cannot be completely removed but only mitigated.

The work to make Secure UPnP method and its usability better continues in future projects.

References

[1] Games@Large. [Online]. Available: http://www.gamesatlarge.eu/.
[2] Y. Tzruya, A. Shani, F. Bellotti, and A. Jurgelionis, "Games@Large – a new platform for ubiquitous gaming," *presented at the Broadband Europe* 2006, Dec 2006, Geneva, Switzerland.
[3] Vesa Pehkonen and Juha Koivisto, "Secure universal plug and play network," *presented at the Information Assurance and Security* 2010 (IAS 2010), August 2010, Atlanta, USA
[4] UPnP Forum. [Online]. Available: http://www.upnp.org/
[5] Digital Living Network Alliance, DLNA, website, http://www.dlna.org/
[6] OpenSSL. [Online]. Available: http://www.openssl.org/