# ARQUITECTURA BÁSICA DE UN NAVEGADOR DVB-HTML PARA MÚLTIPLES TERMINALES


ANEXO C

**José Wilmer Castillo Obando**
**Flavio Andrés Martínez Erazo**

Director
Ing. RODRIGO ALBERTO CERÓN MARTÍNEZ
Asesores
Ing. VICTOR MANUEL MONDRAGÓN MACA
Ing. FRANCO ARTURO URBANO ORDOÑEZ

Universidad del Cauca
**Facultad de Ingeniería Electrónica y Telecomunicaciones**
**Departamento de Telemática**
Línea de investigación: Sistemas telemáticos a la tele-educación
Popayán, Junio de 2009

## TABLA DE CONTENIDO



## INDICE DE FIGURAS



## INDICE DE TABLAS

# ANEXO C: Seguridad en MHP

## C.1 APLICACIONES FIRMADAS

Una aplicación firmada es aquella que está firmada siguiendo los procesos clásicos de claves públicas/privadas, certificados, etc. Las aplicaciones que son firmadas son identificadas con un *application_id* a partir de la gama de aplicaciones firmadas. Las aplicaciones que están sin firmar son identificadas con un *application_id* del rango de aplicaciones sin firmar. Una aplicación con un *application_id* del rango de aplicaciones firmadas, pero que no está firmada es considerada como una autenticación fallida. Una aplicación con una *application_id* del rango de aplicaciones sin firmar es tratada como no firmada, incluso si los archivos pueden ser transmitidos con firmas [1].

### C.1.1 CÓDIGOS HASH

A continuación es descrita la aplicación de códigos de hash a los archivos y directorios. El cálculo de hash considera el contenido y atributos de los objetos en lugar de la información específica de transporte. Como resultado, la autenticación es independiente del protocolo de transporte subyacente. En el caso de una carpeta el valor de hash depende de los valores hash de los objetos vinculados a ella, y así lo prevé un hash de todos los objetos a ser autenticados en el "árbol" debajo de él [2].

### C.1.2 FIRMAS

Los datos autenticados es un sistema jerárquico de archivos (por ejemplo, el DSM-CC OC). La raíz de un árbol autenticado lleva una o más firmas. Esto permite que una o más organizaciones firmar un conjunto de información. La raíz del árbol autenticado puede ser el directorio raíz del sistema de archivos o el directorio principal de un directorio de un subárbol. Las funciones de la firma son [2]:

- Referencia un certificado que contiene la clave pública necesaria para descifrar la firma.
- Identifica el algoritmo hash utilizado.
- El valor de la firma.

### C.1.3 CERTIFICADOS

El certificado proporciona una clave pública que es posible utilizarla para descifrar un código hash contenido en una firma y así permitir a un árbol/subárbol ser verificado. El certificado es firmado por una autoridad de certificación más grande. Para autenticar correctamente un subárbol debe haber una cadena valida de certificados de la firma de un certificado raíz [2].

## C.2 MODELO DE PERMISOS

Gran parte del modelo de seguridad se basa en el de J2SE, por defecto las aplicaciones no podrán acceder a numerosas APIS, lanzando excepciones SecurityException; por ejemplo, las aplicaciones por defecto no pueden leer ciertas propiedades del sistema, usar el canal de retorno o el almacenamiento persistente. Para describir lo realizable, el sistema esta apoyado mayormente en los permisos de los objetos (Permission Objects) de cada API; de esta forma es posible definir con mucho detalle qué funcionalidades son permitidas y a cuáles no [2].

## C.3 SANBOX

Son aplicaciones firmadas o no firmadas que sin un archivo de permisos tienen acceso a todas las APIs para las cuales no hay una señalización de permisos definido. Dicho de otra forma, existen restricciones y servicios accesibles, por defecto que son aplicadas y ofrecidas a las aplicaciones sin firmar y también a las firmadas que no lleven el Permission Request File (PRF) [3].

Los objetos *Permission* son configurados por el entorno para gestionar a nivel de API Java el nivel de acceso a los servicios, lo cual no afecta desde el punto de vista del desarrollador de aplicaciones. Es en el Permission Request File donde el usuario indica (o el broadcaster) qué desea hacer y es mediante la configuración por parte del sistema de objetos Permission como “da permiso” para hacerlo o no. El PRF es almacenado en el mismo directorio en el que reside el XLET y tiene el nombre: dvb.xlet_name.perm [2]

La tabla C.1 muestra algunas diferencias de capacidad de acceso de las aplicaciones firmadas y no firmadas a las APIs que tiene el STB. Estas y otras características las describe en el archivo PRF [4].

| **Característica** | **Aplicaciones NO firmadas** | **Aplicaciones firmadas** |
|---|---|---|
| **Acceso de archivo al PersistentStorage** | No | Si |
| **CA API** | No: org.davic.net.ca.CAModule con los métodos: buyEntitlements, openMessageSession, addMMIListener, queryEntitltements, listEntitlements. | Si |
| **Apps Life Cycle Control** | Sólo las visibles en el APP Listing API provisto por el Service actual para el ServiceContext donde se encuentra la app que desea ejecutarla.<br>• SOLO puede controlar el ciclo de vida de una app que ha lanzado ella: start, pause, stop, resume | Si |
| **Canal de Retorno** | no | si |

| | | |
|---|---|---|
| **(VIA ISP, phone-number)** | | |
| **Tuning API** | no | si |
| **Service Selection** | no | si |
| **User Setting and Preferences access policy** | Solo: User Language, Parental Rating, DefaultFontSize, Country Code. | Todas las preferencias |
| **Network Permissions** | No | Si |
| **Dripfeed** | No | Si |
| **Runtime Code Extension Permission** | SI | Si con PRF, No con PRF |
| **NON-CA smartcard** | No | Si |
| **Cryptographic Service Provider Management** | No | Si |
| **Application Storage** | No | Si |

**Tabla C.1 Diferencias en la capacidad de acceso entre una aplicación no firmada y otra que sí.**

## C.4 PROCESO DE FIRMADO DE UNA APLICACIÓN

A continuación es descrito el proceso de firmado de una aplicación cualquiera, que integra todo lo necesario para que una aplicación cumpla con la especificación MHP [2].

### C.4.1 Archivos HASH

Una aplicación MHP contiene archivos, imágenes y carpetas ordenadas en forma de árbol. Para cada uno de los archivos de una carpeta X, es generada una cadena de caracteres llamada Digest (algoritmo MD5 o SHA-1) y es incluido en un archivo dvb.hashfile en la misma. Mediante la comparación del Digest con el fichero sería posible saber si el fichero ha cambiado o no.

De igual forma es para la carpeta que contiene a la carpeta X, el archivo dvb.hashfile tiene los Digest de los archivos, además del Digest final de la carpeta X y de otras carpetas contenidas si las hubiesen.

El proceso sigue de igual forma hasta el directorio raíz, donde tendrá el dvb.hashfile de mayor nivel. Los archivos Hash deben contener la lista de todos los componentes de la carpeta donde permanecen con su respectivo Digest**.** Es posible no tener que "codificar" absolutamente todo, es posible dejar fuera de la encriptación ficheros de datos o imágenes. Además, todos los elementos deben estar referenciados dentro del Hashfile.
Es posible generar los Digest para grupos de archivos, en lugar de uno por archivo, sin embargo, no es recomendable, porque para acceder solo a un archivo es necesario abrir y comprobar todos los archivos.

### C.4.2 Firmas de archivos

El paso luego de disponer un dvb.hashfile y en él un Digest de toda la aplicación, es firmarla. Para esto es generado el Digest del dvb.hashfile raíz y firmado, es decir, el operador de Red lo encripta usando su clave Privada aplicando el algoritmo RSA, y guarda el resultado en el fichero dvb.signature.id, donde id empieza en 1 y sirve para identificar posibles firmantes diferentes. Este fichero *dvb.signature.id* reside en el mismo directorio que el dvb.hasfile raíz.

### C.4.3 Certificados

Teniendo los archivos hash, además del hash de toda la aplicación. El hash principal es firmado con una llave privada que sólo es posible abrir con una llave pública. La llave publica es incluida en un certificado en el directorio raíz, que asegura que esa clave pertenece al operador de red firmante correcto. Este certificado reside en un archivo en el directorio raíz llamado *dvb.certificate.id***,** donde id empieza en 1 y tiene el mismo valor que el *dvb.signature.id* correspondiente. El tipo de certificado usado es una variante de Internet Profile de X.509**.**

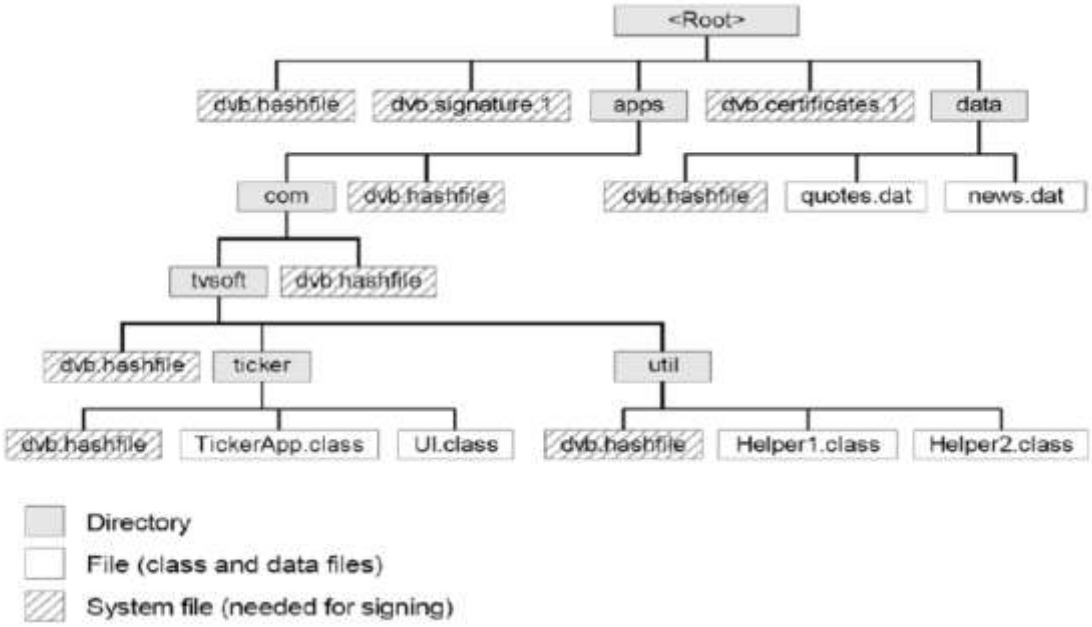


**Figura C.1 Ejemplo de una aplicación firmada**

# BIBLIOGRAFIA

[1] S. Morris, A. Smith-Chaigneau, "*Interactive TV Standards*", Editorial Focal Press, 2005.

[2] DVB, "*Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.1.3".*

[3] The MHP Knowledge Project (MHP-KDB), "*The MHP-Guide, A comprehensive Guide to the Multimedia Home Platform, the underlying technology and possible uses*", 2006, disponible en: http://www.mhpkdb.org/publ/mhp-guide.pdf

[4] E. M. Schwalb, "*iTV Handbook: Technologies and Standards*", Editorial Prentice Hall, 2003.