Chapter 5 Multimedia Content Guard Application Based on MPEG-4 IPMP-X

This chapter, we will introduce implementation of two multimedia content guard application based on MPEG-4 IPMP-X. The first application is Digital TV Conditional Access,we will describe the scenario which is the general model of Digital TV Condition Access. After we introduce the scenario, we will describe how to work based on MPEG-4 IPMP-X. The second application is the DVD copy protection. It is described the same ways as the first application. The last section, we trace a software reference code supported by IM1.The software reference code is about MPEG-21 Right Expression Language in a MPEG-4 IPMP-X. We will explain this example in the last section.

5.1 Application 1-Digital TV Conditional Access

In general, conditional access systems that are associated with digital entertainment services or broadcast data services are typically put into use to enforce the business arrangements between content provider, service provider, and subscriber. To protect content, it is digitally scrambled at the distribution point. To descramble the content, secret keys for descrambling are sent to authorized terminals where the content is decrypted so that it can subsequently be decoded and displayed on a TV set.

In conclusion, the conditional access system is responsible for creating the control messages, encrypting the content, and delivering the messages to the terminal. Various suppliers of conditional access systems have different methods for generating and delivering encryption keys, different terminals use different techniques and implementations for protecting and using the keys to decrypt the content.

5.1.1 The Digital TV Conditional Access Model

In general, the terminal is considered as a set-top box. Protected content arrives at the set-top box as a series of streams, the scrambled program.

Within the set-top there is usually a special processor that performs the function of decrypting the key. Various set-top box designs the Condition Access Module in different physical location. The following lists some of these models.

(1) Embedded within the set-top: All of the security elements are internal to the set-top box the security circuits are an integral part of the set-top electronics. The model is illustrated in figure 5-1.

(2) Split between the set-top box and removable card: The removable card is called a Smart Card. Another security circuit and firmware are embedded in the set-top to decrypt the content when it receives the messages from the Smart Card. The model is illustrated in figure 5-2.

(3) Contained completely on a removable module:The removable module(called Common Interface Module or “CIM” in DVB region or “POD”)receives the entire transport stream and performs message and content decryption-then sends the decrypted content into the set-top box. The model is illustrated in figure 5-3.

The three models represent different levels of content security and cost. The completely embedded approach is the most secure, since neither the content nor the control messages leave the set-top box. This is also the least expensive since external connectors and removable elements are not required. The Smart Card system is less secure because the decrypted control messages travel across an open interface between the Smart Card and the set-top where they can be intercepted and analyzed. Lastly, the completely removable security system also has security vulnerabilities and

is the most expensive implementation.

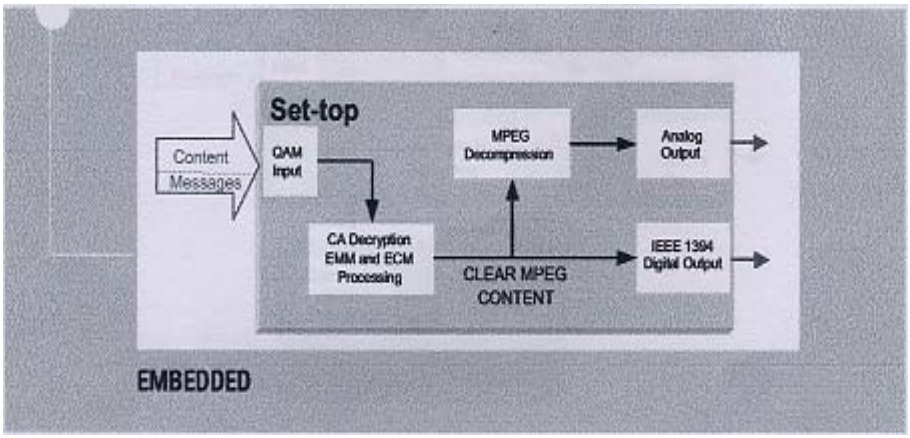


Figure 5-1 Embedded within the Set-top Box for CA Module, Motorola

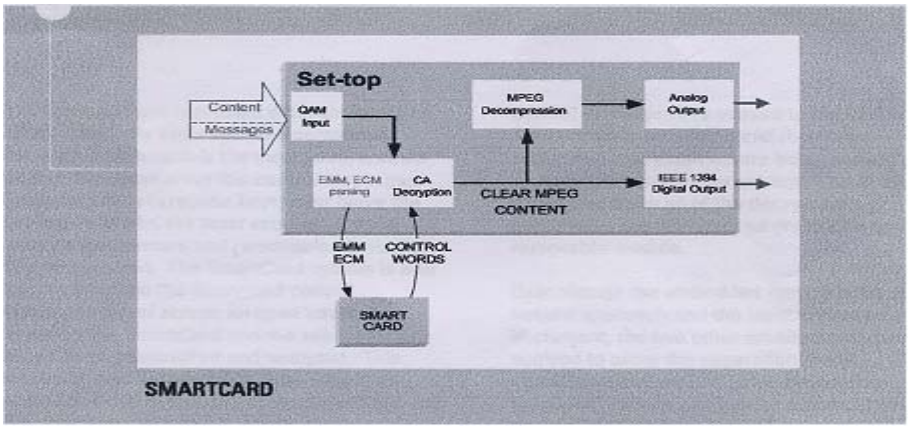


Figure 5-2 Split between the Set-Top and Removable Card for CA Module, Motorola

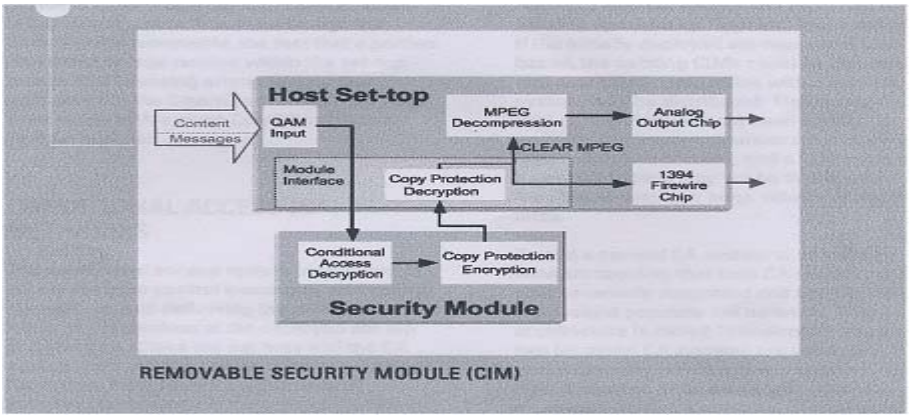


Figure 5-3 Completely on a Removable Module for CA Module, Motorola

5.1.2 The Description of Conditional Access Application Based On MPEG-4 IPMP-X

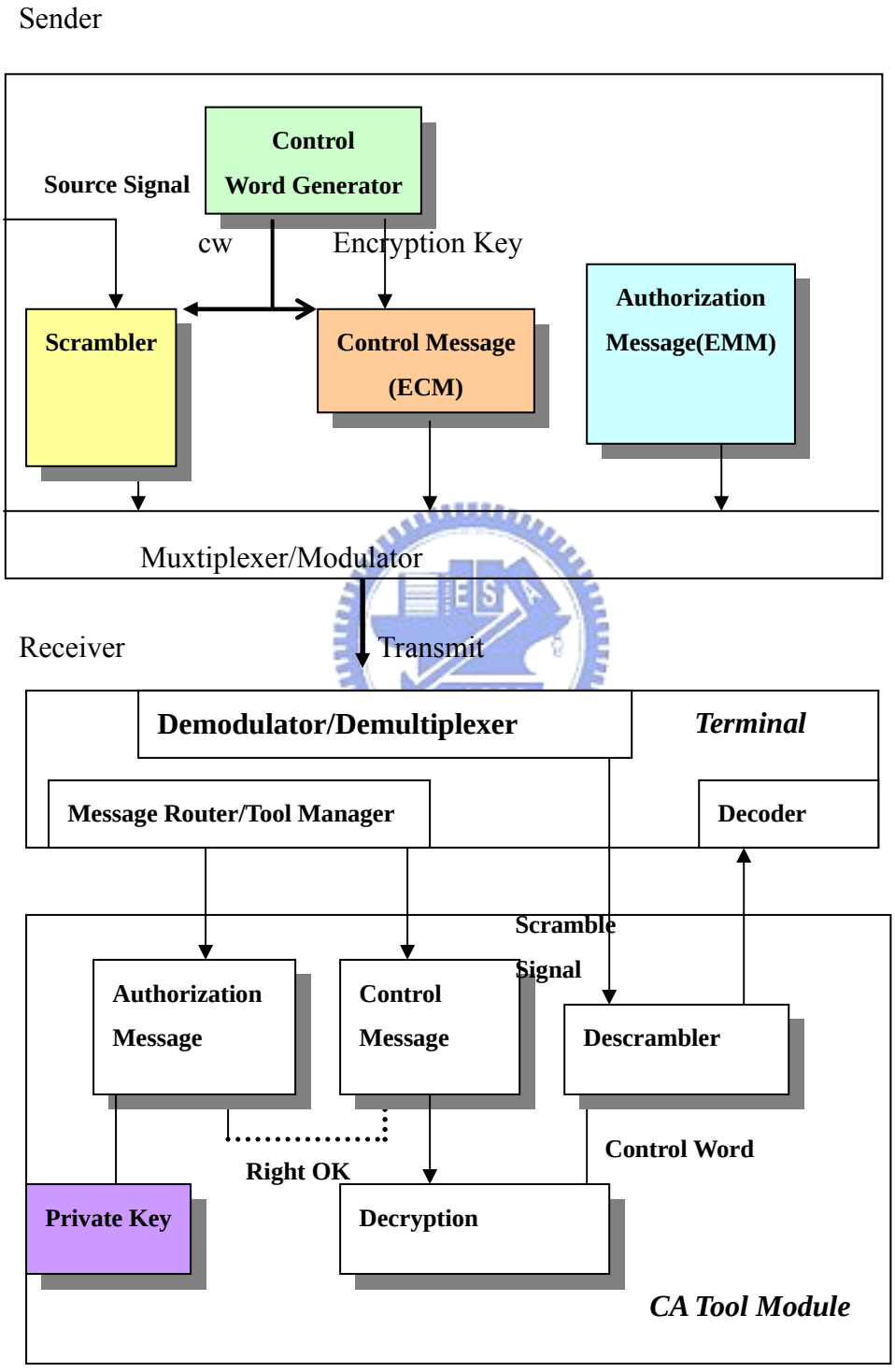


Figure 5-4 Conditional Access Application Scenario Based On MPEG-4 IPMP-X

The figure 5-4 is general model of digital TV Conditional Access. We can easily map the various design models into figure 5-4. ***The Scenario does not consider multiple CA system (e.g MultiCrypt , SimulCrypt).***

We follow scenario with introduction of term definition of Digital TV Conditional Access.

(1) Control Word(CW):

A packet containing the secret information for the scrambling algorithm.

(2) Entitlement Control Message(ECM):

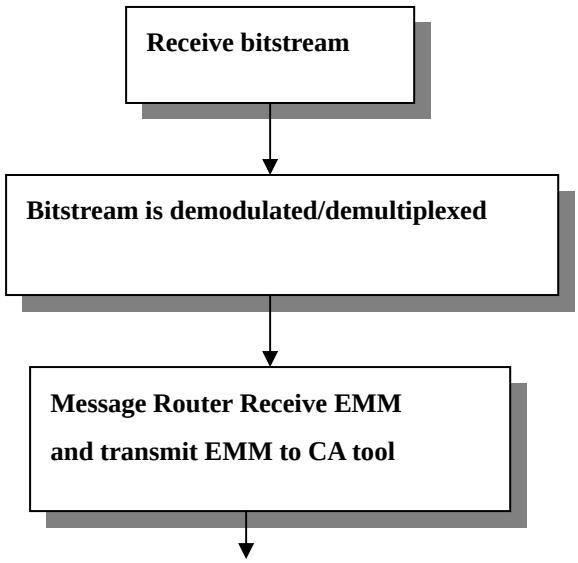
It carries an encrypted form of control words(CW) in the periodic time. ie an identification of the service and of the conditions required for accessing this service.

(2) Entitlement Management Message(EMM):

It carries entitlements or keys to users ,or to invalidate or delete entitlements or keys.

5.1.3 Flow Description of Digital TV Conditional Access

From figure 5-4 we will make up the flow of the receiver in the following.



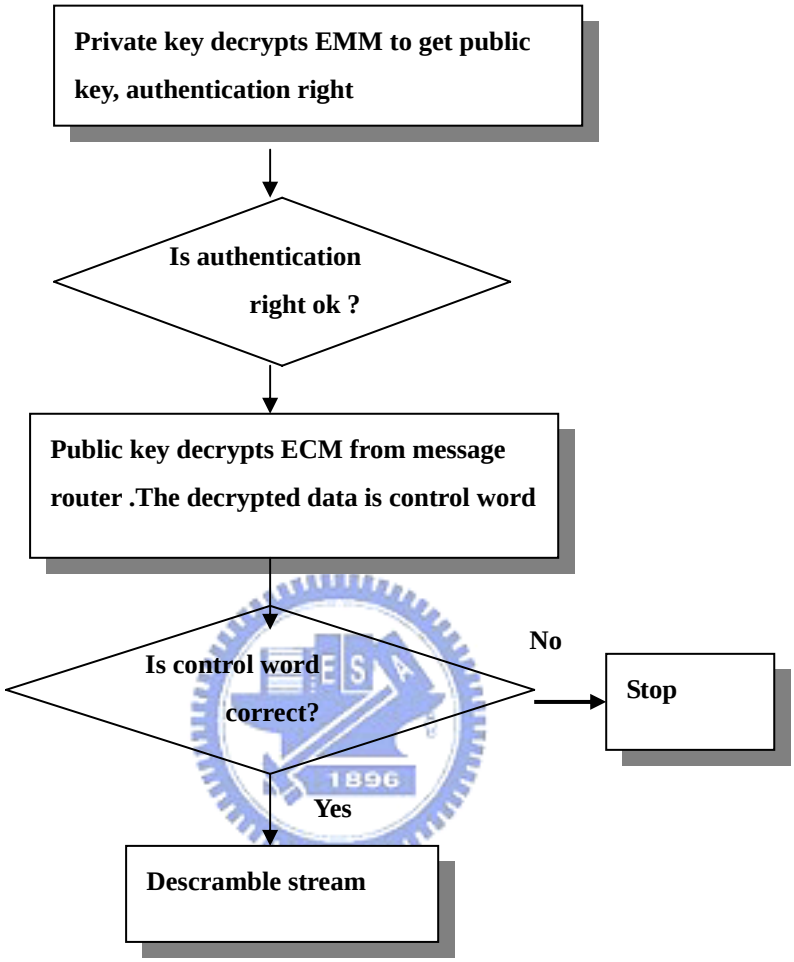


Figure 5-5 A Flow of Conditional Access Application Scenario Based On MPEG-4 IPMP-X

5.1.4 Algorithm Description of Digital TV Conditional Access

We use security technique including RSA algorithm [20] and Perfect Hashing [18][29][30].

Firstly, we will introduce RSA algorithm in general case.

(1) Key Generation Algorithm

(a)Generate two large random primes, p and q, of approximately equal size such that their product n = pq is of the required bit length, e.g. 1024 bits.

Compute n = pq and phi = (p-1)(q-1).

Choose an integer e, 1 < e < phi, such that gcd(e, phi) = 1

Compute the secret exponent d, 1 < d < phi, such that

ed $\equiv$ 1 (mod phi).

(b)The public key is (n, e) and the private key is (n, d). The values of p, q,

and phi should also be kept secret.

n is known as the modulus.

e is known as the public exponent or encryption exponent.

d is known as the secret exponent or decryption exponent.

(2) Encryption/Decryption

The following is a pair key (n, e) and (n, d) for sender/receiver on RSA algorithm flow.

Sender A does the following:

(a)Uses key (n, e).

(b)Represents the plaintext message as a positive integer m < n.

(c)Computes the ciphertext c = m^e mod n.

(d)Sends the ciphertext c to B.

Recipient B does the following:

(a)Uses key (n, d) to compute m = c^d mod n.

(b)Extracts the plaintext from the integer representative m.

We introduce RSA algorithm and then use RSA algorithm on Digital TV Conditional Access, the figure 5-6 describes our application using RSA algorithm.

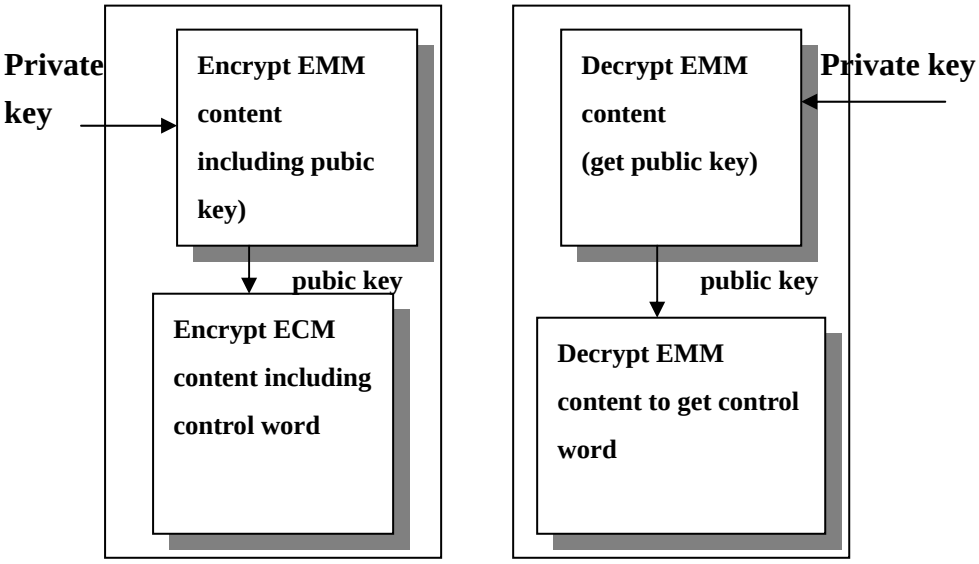


Figure 5-6 Encrypt / Decrypt of Conditional Access Application Scenario

The next paragraph, we will use concept of Public-key Cryptosystems [20] to implement our application.

We use an asymmetric cryptosystem for encryption and decryption. The basic principles of a public-key cryptosystem are described in figure 5-7. It shows a sender S and receiver R communicating via an insecure communication channel. Before sending any plain text P to R, the sender S first encrypts it , use S's private key, K_S^{priv}, and a second time using R's public key, K_R^{publ}. The receiver decrypts the received message $E(E(P, K_S^{priv}), K_R^{publ})$ first using R's private key and S's public key, getting the original message P. Public-key cryptosystem offer many advantages ,including enforcing secrecy, integrity and authenticity.

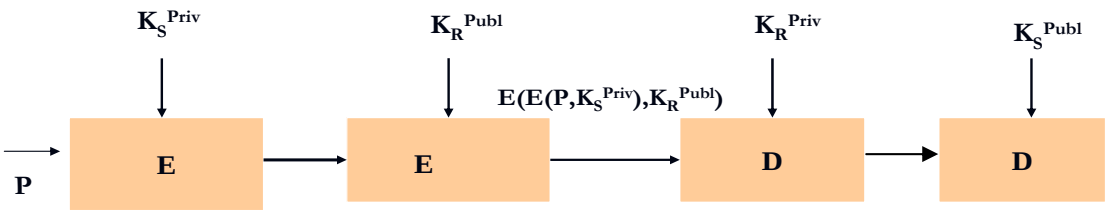


Figure 5-7 Principles of public-key cryptography

On the other hand, we also use Perfect Hashing[18][29][30] to check whether Control Word is correct. The Control Word is send by the sender. Perfect hashing guarantees that you get no collisions at all. It is possible when you know exactly what set of keys you are going to be hashing when you design your hash function.

Perfect hash algorithm uses an initial hash to find a pair (A,B) for each keyword, then it generates a mapping table tab[] so that A^tab[B] is unique for each keyword. The size of tab[] is always a power of two. Finding values for tab[] such that A^tab[B] causes no collisions is known as the "sparse matrix compression problem", which is NP complete. Like most NP complete problems, there are fast heuristics for getting reasonable (but not optimal) solutions. The heuristic for this paper uses spanning trees .We uses this algorithm to check whether control word is correct. Figure 5-8 is a Flow of Control Word for Descramble

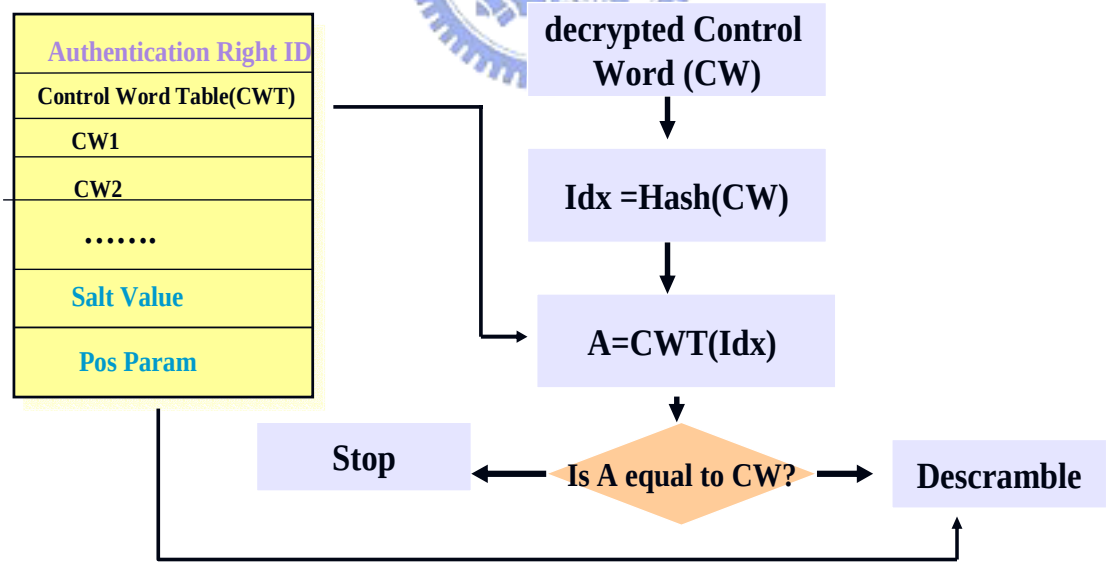


Figure 5-8 A Flow of Control Word for Descramble

Comment: For the sake of safety, the provider must update the control word table for a period time.

Scrambling algorithm

Lastly, we introduce scrambling algorithm. We uses the simplest way to implements scrambling algorithm. After hash table is built up, the salt value is created We will put salt value in the video stream for each n-byte position. The n-byte is opaque data carried in EMM message.

5.1.5 Implementation of Conditional Access Application Scenario Based On MPEG-4 IPMP-X

5.1.5.1 Trif file for Conditional Access Application Scenario

The IPMP-X of IM1 system is standalone. We develop application program based on this. The sender is simulated by reading the trif file. We list the important part of *.txt and *.scr for this application in the following and comment . As for how to create a trif file, please refer to section 4.1.

(1)*.scr:

Scheme 5A:

```
InitialObjectDescriptor {
// initial object description
….
toolListDescr{ //toollist description
ipmpTools [
            IPMP_Tool {
            IPMP_ToolID 111 // Tool id 111
            isAltGroup FALSE
            }
          ]
    }
esDescr [
    ES_Descriptor {
    // OD Stream description
    }
    ES_Descriptor {
    // BIF Stream description
```

```
        }
        ]
}

{ // Object description
        ObjectDescriptorID 13
        esDescr [
                {
                // Audio stream description
                }
        ]
}

{
        ObjectDescriptorID 12
        esDescr {
                //video object description
                }
                ipmpDescrPtr [
                // The IPMP_DescriptorPointer appears in the ipmpDesPtr section
                //of an video object. The presence of this descriptor indicates that //the stream
                associated with this descriptor is subject to protection //and management by the
                IPMP Tool specified in the referenced //IPMP_Descriptor or
                IPMP_ToolDescriptor.
                                {IPMP_DescriptorID              0xff
                                IPMP_DescriptorIDEx     69 //ToolDescriptorID
                                IPMP_ES_ID                      0x00
                                }
                ]
        }
}
```

(2) *.trif :

Scheme 5B:

```
DEF N32 Group {
// description of shape ,position and appearance
```

```
}
// IPMP_ToolDescriptor carries IPMP information for IPMP Tool 111
// Instance
UPDATE IPMPDX [
        IPMP_Descriptor {
                IPMP_DescriptorID       0xff
                IPMPS_Type              0xffff
                IPMP_DescriptorIDEx     69
                IPMP_ToolID             111
                controlPointCode        0x01
                sequenceCode            0x0
                IPMPX_Data [
                        // EMM message is carried in the OpaqeData. The
                        // following is decrypted EMM data.
                        IPMP_OpaqueData{
                        //auth id -> public key n-> public key -> scramble pos gap-> how
                        many bytes are scrambled in each pos gap.
                        opaqueData
                        "27#43?1a8^99!b4?96=c3!fb:c3?4a<c3:91?91=c3:c3&c3!cc&96:d7?1
                        22"
                        }
                    ]
                }
        ]
…..
// IPMP_ToolDescriptors are conveyed in update command. The ECM message
// is sent for each 0.1 sec
At 500 UPDATE IPMPDX [
        IPMP_Descriptor {
                …….
                IPMPX_Data [
                        / /ECM message is carried in the OpaqeData. The
                        following is decrypted ECM data.
                        IPMP_OpaqueData{
                            opaqueData "1b8!196:48*93=106<135<48?135?ab?ab&1e6"
        }
```

```
            ]
        }
]
…..
At 1500 UPDATE IPMPDX [
        ….
]
At 2500 UPDATE IPMPDX[
…
]…
```

5.1.5.2 The description of Implementation of Conditional Access Application Based on MPEG-4 IPMP-X

We create a tool for CA module. Description of the tool module is described in section 4.3.5. The major code of our CA application program is to expand the ProcessData() in scheme 5C and ProcessMessage() in scheme 5D in the tool framework. We list the important code and comment .

Scheme 5C:

```
bool    IPMP_CATool::ProcessMessage ( ToolMessage base
                                        , IPMP_Data_BaseClass    *msg )
{
   ….//The tool receives message. The message stores in RcvData array.
      int b = msg->GetTag();
       switch (msg->GetTag()){
            case(TAG_IPMP_OpaqueData):
                 {
                  ….
                  // decrypt and parse ECM message
                     CWOpaqueParser(RcvData);
                              …
                  // check control word whether is correct
                     if ( strcmp(Input[InStr.hash].kname,InStr.kname)!=0)
                           CWFlag = FALSE;
                     else CWFlag = TRUE;
                     if (CWFlag != TRUE)
```

```
                    DeScrambleFlag = FALSE;
                ….;
                }
            …..
            case(TAG_User_Initialize):
                {
                …
                //ReceiveMessage() receives IPMP message
                // Parse message

                {
                switch (..){
                    case (TAG_IPMP_OpaqueData):
                        {
                        ….
                        //Decrypt and parse EMM message
                        EMMOpaqueParser(RcvData);
                        }
                    }
                }
                //The tool will transfer message to the terminal.
              m_pMRInterface->ReceiveMessage(&Msg);
                return true;
                }
            }
        return false;
}
```

Scheme 5D:

```
bool IPMP_CATool::ProcessData (LPBYTE pInput
                                    , int          nInputLength
                                    , DWORD dwTime ))
{
// If DeScrambleFlag is equal to true, it must meet all conditions described in
section 5.1.2.2

   if( DeScrambleFlag == TRUE)
   {…
```

```
      DeScramble(ProtectData,nInputLength);
    …
 }
 return m_pNextTool->ProcessData( pInput, nInputLength, dwTime );
}
```

Scheme 5C and scheme 5D are the major parts of Tool Module of Digital Conditional Access.

5.1.6 Demo for Conditional Access Application Scenario

The following demo shows three different cases. Note that control word is sent for each 0.1sec period..

Figure 5-9 shows that the user has not authentication right such that the media stream can not be descrambled.

Figure 5-10 shows that the user has authentication right and the control words are right such that the median stream can be descrambled.

Figure 5-11 shows that the user has authentication right but the control words are not right such that the media stream can not be kept on scrambling when the amount of money of user has expired.

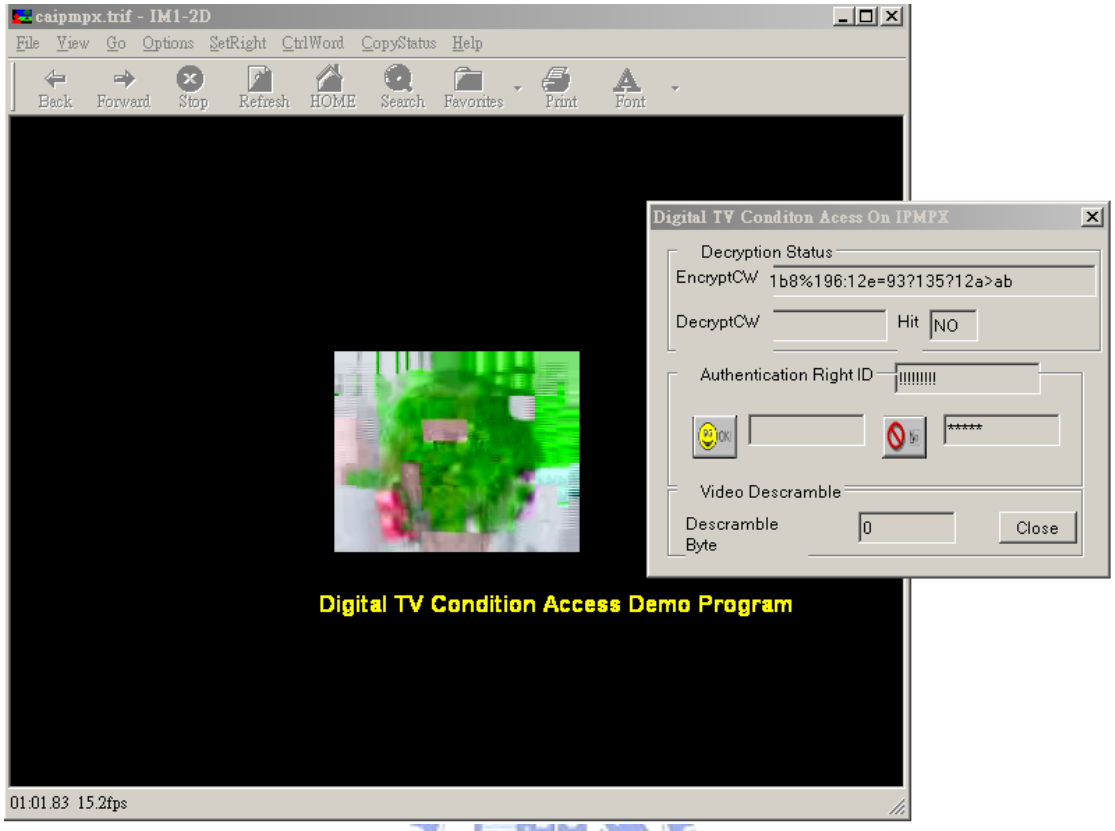


Figure 5-9 Demo 1 for Digital TV Conditional Access

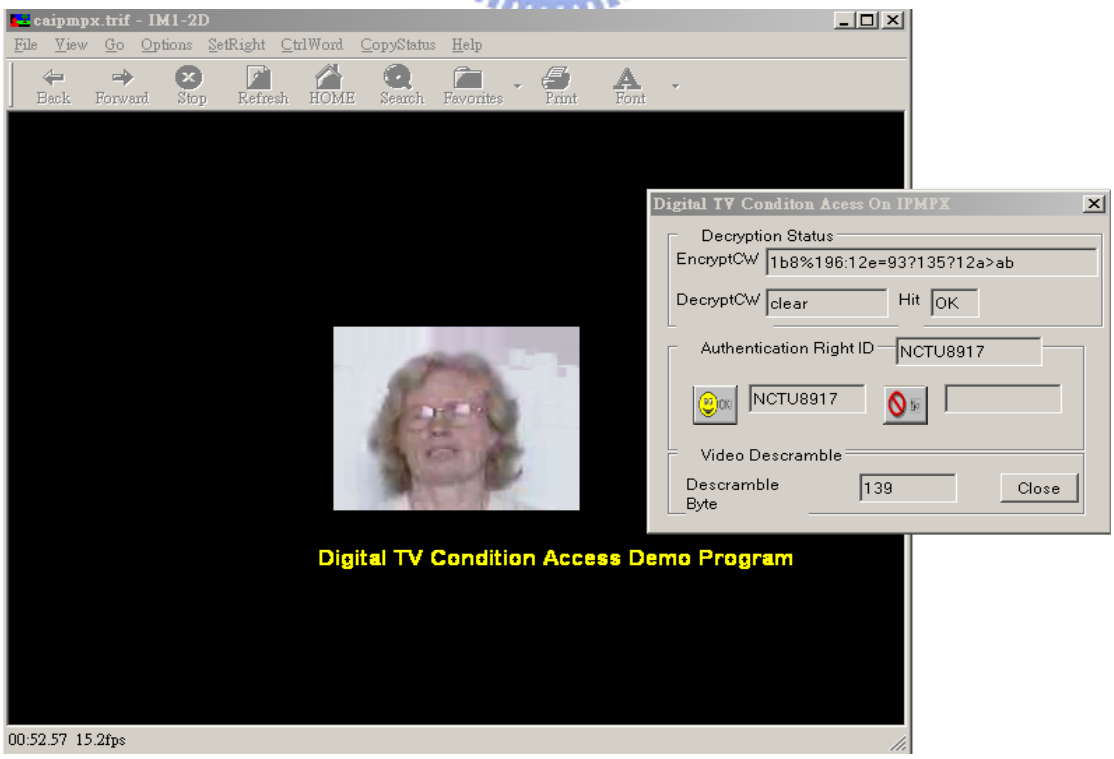


Figure 5-10 Demo 2 for Digital TV Conditional Access

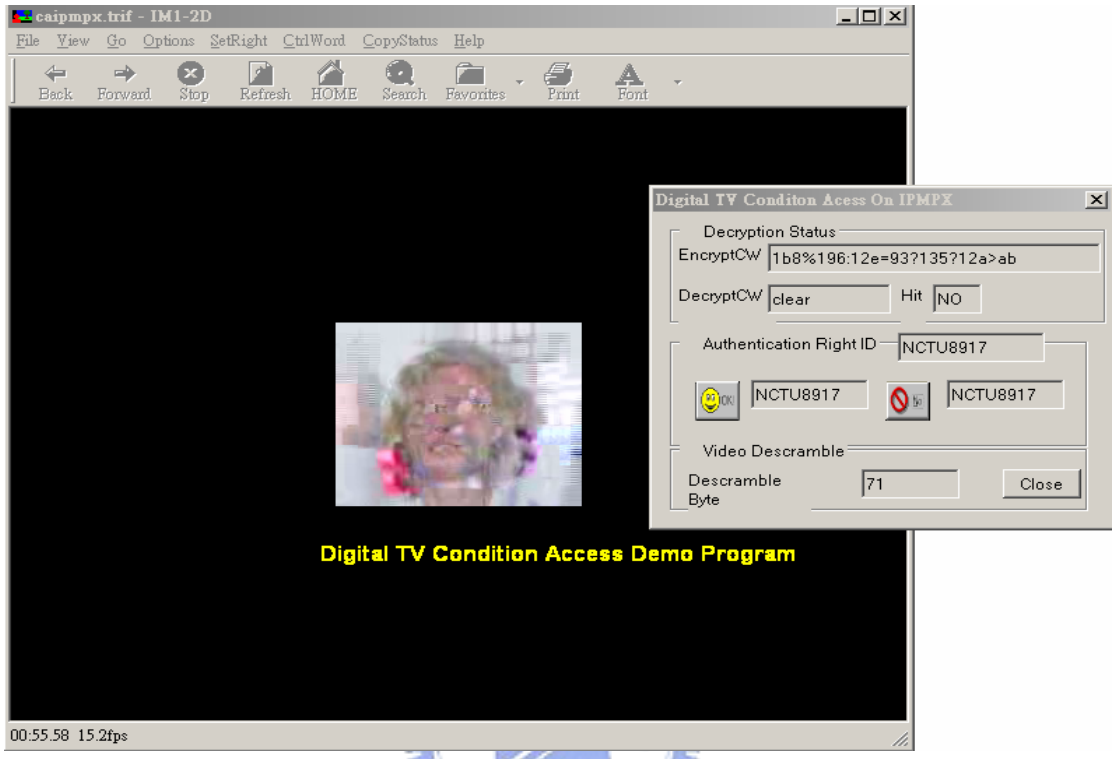


Figure 5-11 Demo 3 for Digital TV Conditional Access

5.2 Application 2- DVD Copy protection

The prospect of consumer digital versatile disk (DVD) recorders highlights the challenge of protecting copyrighted video content from piracy. The copy protection system broadly tries to prevent illicit copies from being made from either the analog or digital I/O channels of DVD recorders. Several watermark applications that are currently in place or very close to being released were discussed. We focus on the watermark application on DVD copy protection.

[10]Since 1996, digital watermark has been considered on of the potential "safety net"for copy protection and playback control of the DVD contents. A technical subgroup called Data Hiding Subgroup was formed in June 1997 under the DVD CPTWG to evaluate the technical feasibility of a total of eleven watermark technology proposals. The result showed that the current watermark technology has potential technical feasibility to meet the technical requirements for the DVD copy control application.

Watermarking technology can be viewed as a way to provide a secure data channel along with the contents without modifying the installed-base Consumer Electronics device. The embedded watermark is transparently passing through the conventional data path, and will only be detected at the digital recorder. When the watermark detection is mandated in these recorders, this watermark can be used to trigger the copy protection mechanism implemented in it.

[10]In this application, the data called Copy Control Information (CCI) is embedded into the video data to indicate that the status of the contents is "Never Copy","One Copy Allowed"or "Copy Freely".Recording devices will be mandated to facilitate a "watermark detector" to detect the embedded CCI from the incoming and outgoing media stream, and responding properly to recording/playback rules that are

defined.

5.2.1 Detector Placement

In the scenario of Figure 5-12, the watermark detector is located in the DVD drive. The scenario places the watermark detector in the DVD drive. This has the obvious advantage that as long as the watermark is not compromised, pirated content will never leave the drive(in playback mode) or will never get copied onto a disk(in recording mode).Given that assumption, the location of the watermark detector in the drive is secure and tamper resistant. Record control will prevent noncompliant MPEG bit stream from being recorded.

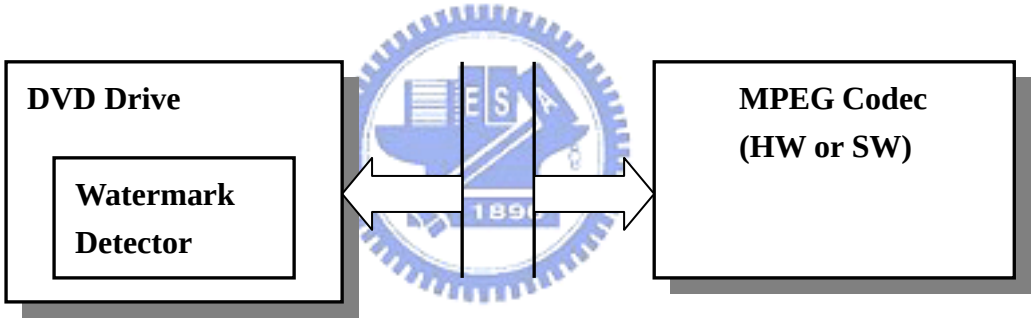


Figure 5-12 Watermark Detector Placement

5.2.2 Description of DVD Copy protection Scenario Based on MPEG-4 IPMP-X

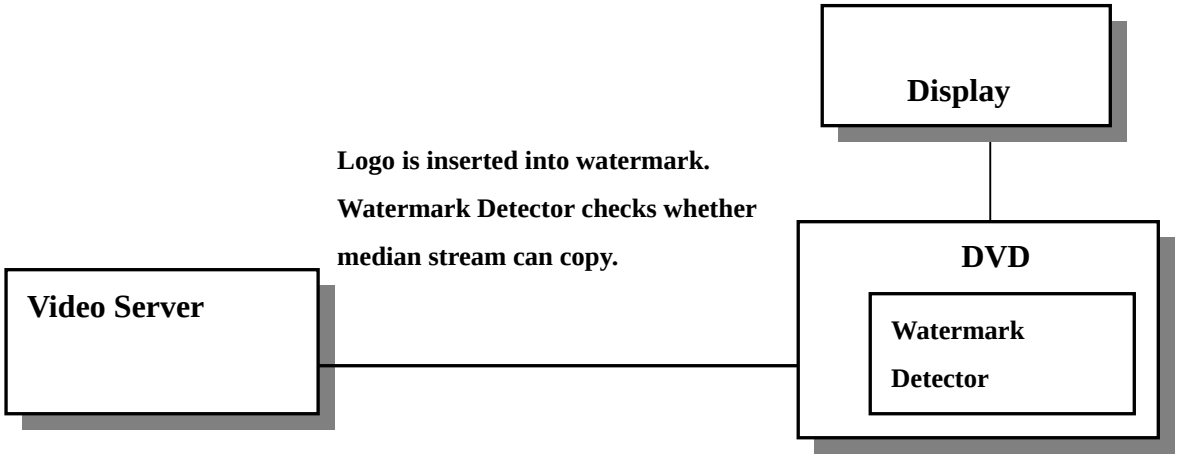


Figure 5-13 DVD Copy Protection Scenario Based on MPEG-4 IPMP-X

We map the figure 5-13 into MPEG 4 IPMP-X terminal architecture, as shown in figure 5-14.In this figure, the IPMP Message data and protected data is filtered to watermark detector tool. Then, the watermark detector checks whether the watermark information can copy.

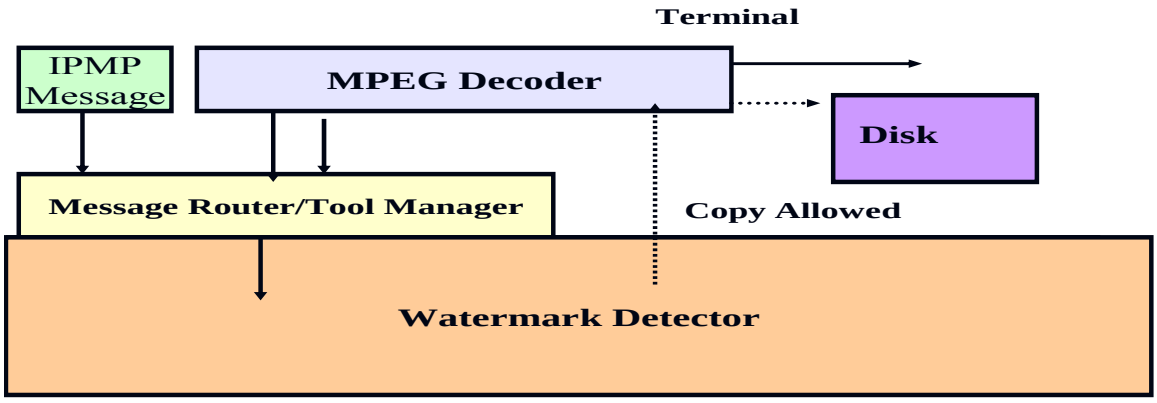


Figure 5-14 DVD Copy Protection Scenario on MPEG-4 IPMP-X Terminal Architecture

5.2.3 Flow Description of DVD Copy Protection

In Figure 5-15, we describe DVD Copy protection Control Flow for Watermark Encoder

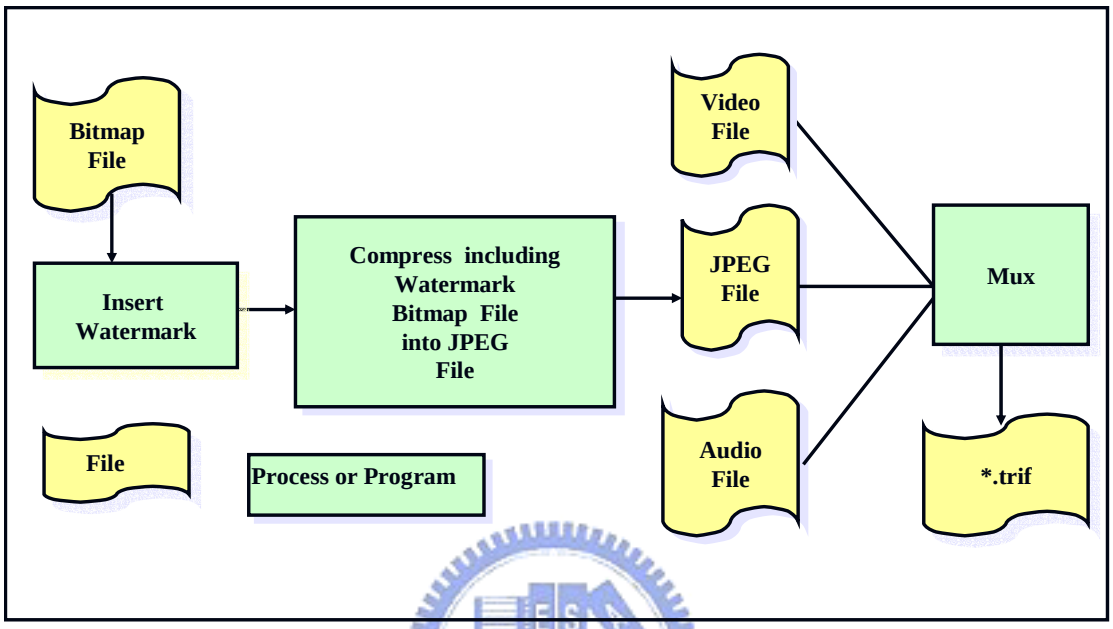


Figure 5-15 Watermark Encoder Control Flow

In Figure 5-16,we describe DVD Copy protection Control Flow for Watermark Detector

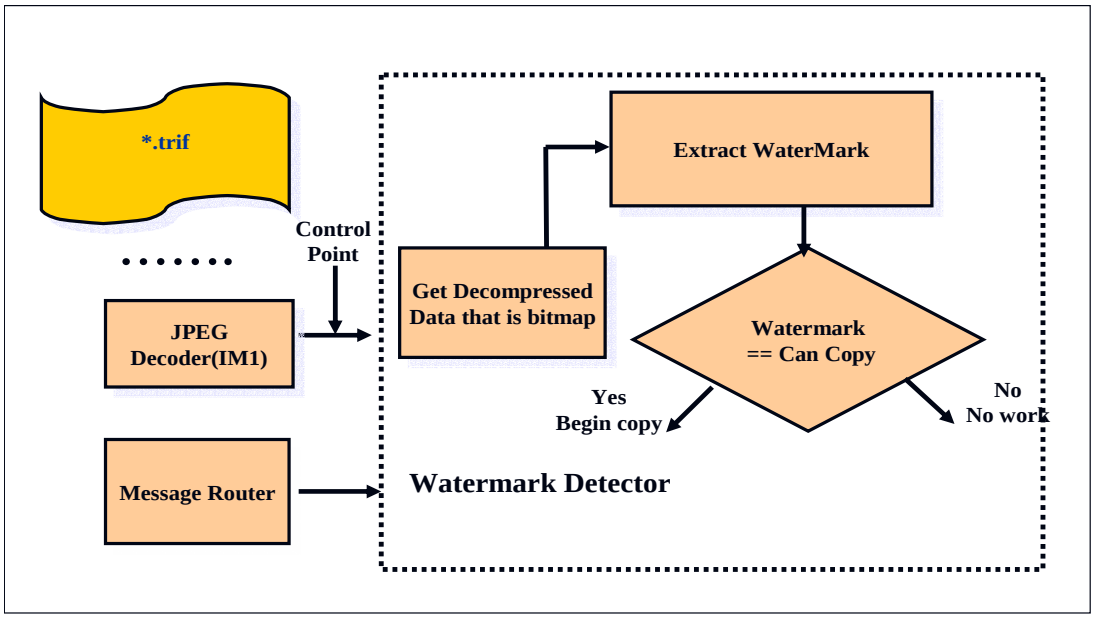


Figure 5-16 Watermark Decoder Control Flow

5.2.4 Watermark Algorithm Description [12]

Our watermarking algorithm uses code division to embed integer message, represented as sequences of 8 bits. The mapping of a message into a sequence of symbols is a mapping of the message into a bit string. To modulate a message, the i^{th} bit of the string is then mapped into $W_{AL}[i,1]$ if it is 1,or $W_{AL}[i,0]$ if it is 0. Let $W_{AL}[i,s]$ correspond to symbol s at location i.(s is either 0 or 1).

The two reference patterns for a given location must be maximally distinguishable from each other. Therefore, we can construct an optimal binary system by designing a single reference pattern, W_{ri}, for each bit location, i , and using that pattern to encode a 1, or the negation of that pattern to encode a 0. In other words, for any bit location, i , $W_{AL}[i,1] = w_{ri}$ and $W_{AL}[i,1] = -w_{ri.}$

Each of the base reference patterns,W_{r1}, W_{r2}, …,W_{r8} ,is generated pseudo randomly according to a given seed.

The message pattern ,W_m , that encodes a given message ,m , expressed as sequence of bits ,m[1], m[2]…m[s], is given by

(1). Insert watermark:

$W_{mi} =$ W_{ri} if m[i] = 1

$-W_{ri}$ if m[i] = 0

$W_{tmp} = \Sigma W_{mi}$

$W_m = W_{tmp} / S_{Wtmp}$ S_{Wtmp}: standard deviation

$W_{m:}$ unit standard deviation

$C_w = C_0 + \alpha W_m$ Where C_0 is the original image, and α is a strength parameter input

(2) Extract watermark

$Z(C_w, W_{ri}) = C_w \cdot W_{ri} > 0$ then $m_i = 1$

else $m_i = 0C_w$: watermark image

W_{ri} : reference patterns, W_{ri} is generated pseudo randomly. The seed value is the same as that of inserting watermark.

(3) We convert Bitmap RGB into YCrCb and watermark inserts only into Y.

(4) In our application, after we insert watermark into bitmap file, the watermarked file is compressed into JPEG file. When we extract the watermark value from the decompressed data, we find that the watermark value is usually not correct. It can be solved to add the watermark to bitmap several times.

5.2.5 An Implementation of DVD Copy Protection Application on MPEG-4 IPMP-X

5.2.5.1 Trif file for DVD Copy protection Application Scenario

The sender is simulated by reading the trif file. We list the important part of *.txt in scheme 5F and *.scr in scheme 5E for this application in the following and comment .As for how to create a trif file, please refer to section 4.1.

(1) *.scr:

Scheme 5E:

```
InitialObjectDescriptor {
// initial object description
….
}
ObjectDescriptor {
ObjectDescriptorID    11
esDescr {
        //JPEG file
        ES_ID 9
        ….
        ipmpDescrPtr [
                // The IPMP_DescriptorPointer appears in the ipmpDesPtr section
                //of an JPEG object.
                        {       IPMP_DescriptorID           0xff
```

```
                    IPMP_DescriptorIDEx     11
                    IPMP_ES_ID              9
                }
        ]
    }
}
{   ObjectDescriptorID 13
        esDescr [
            {
            // Audio stream description
            }
        ]
    }
{
    ObjectDescriptorID 12
        esDescr {
            //video object description
            }
            ipmpDescrPtr [
            // The IPMP_DescriptorPointer appears in the ipmpDesPtr section
            //of an video object.
                        {
                        IPMP_DescriptorID       0xff
                        IPMP_DescriptorIDEx     69
                        IPMP_ES_ID              0x2115
                        }
            ]
        }
    }
```

(2) *.trif:

Scheme 5F:

```
DEF N32 Group {
    // description of shape ,position and appearance
    }
// The first update command shows that a tool is instantiated , the tool 111
//performs its function on JPEG object between the decoder and composition
```

```
//buffer and IPMP_ToolDescriptors transmits watermark related information by
//IPMPX Data Class.

UPDATE IPMPDX [
    IPMP_Descriptor {
        IPMP_DescriptorID       0xff
        IPMPS_Type              0xffff
        IPMP_DescriptorIDEx     11
        IPMP_ToolID             111
        //control point = 2, between the decoder and the composition buffer
        controlPointCode        0x2
        sequenceCode            0x1
        IPMPX_Data [
            IPMP_VideoWatermarkingInit{
                                inputFormat     0x6C //JPEG
                                requiredOp      1    // extract watemark
                                hasOpaqueData   1
                                 //high ->width ->seed
                                opaqueData      "104:104?65535"
      }
        ]
    }
]
//The second update command shows that the tool 111 performs its function on    //video object
between the decoder buffer and the decoder and the tool updates //tool instantiation.
UPDATE IPMPDX [
    IPMP_Descriptor {
        IPMP_DescriptorID       0xff
        IPMPS_Type              0xffff
        IPMP_DescriptorIDEx     69
        IPMP_ToolID             111
        controlPointCode        0x01
        sequenceCode            0x0
        IPMPX_Data [
            ….
            }
        ]
```

```
        }
]

// Mux JPEG ,Video ,Audio
UPDATE OD [
        {objectDescriptorID    11
        muxScript watermark.scr
        }
        {objectDescriptorID    12
        muxScript watermark.scr
        }
        {objectDescriptorID    13
        muxScript watermark.scr
        }
```

5.2.5.2 The Description of Implementation of DVD Copy Protection Application Based on MPEG-4 IPMP-X

We create a tool for watermark detector module. The major code of our DVD Copy protection application program is to expand the ProcessData() in scheme 5G and ProcessMessage() in scheme 5H in the tool framework. We list the important code and comment

Scheme 5G:.

```
bool IPMP_CATool::ProcessMessage ( ToolMessage base
                                        , IPMP_Data_BaseClass    *msg )

{
   …..//The tool receives message. The message stores in RcvData array.
      int b = msg->GetTag();
       switch (msg->GetTag()){
             case(TAG_IPMP_OpaqueData):
                   {
                   }
             …..
             case(TAG_User_Initialize):
```

```
                {
                …
                //ReceiveMessage() receives IPMP message
                {
                switch (..){
                        ….
                        case (TAG_IPMP_VideoWatermarkingInit):
                                        {
                                        ….
                // Parsing IPMP message from sender
                                        WatermarkDataParser();
                                        break;

                                        }
                                }
                        }
                        //The tool will transfer message to the terminal.
                    m_pMRInterface->ReceiveMessage(&Msg);
                        return true;
                        }
                }
        return false;
}
```

Scheme 5H:

```
bool IPMP_CATool::ProcessData (LPBYTE pInput
                                        , int           nInputLength
                                        , DWORD dwTime ))
{

        if      ( FirstAccessUnitFlag)          //JPEG Access Unit
                {
                ….
                // pInput is decompressed JPEG data. This tool extracts watermark for //this data.
                memcpy(ProtectData, pInput,nInputLength );
                ProtectDataLen = nInputLength ;
                //Call Watermark Extract
```

```
            ExtractMark    = ExtractWaterMark();
            return(m_pNextTool->ProcessData( pInput, nInputLength, dwTime ));
            }
            //After watermark extracts , it shows this stream can be copied.
      if    (ExtractMark == CopyRightKey )
            {
           MatchFlag = true ;
            return(m_pNextTool->ProcessData( pInput, nInputLength, dwTime ));
            }
   else return false;
}
```

5.2.6 Demo for DVD Copy Protection Application Scenario

The following demo shows two different cases.

Figure 5-17 shows that the watermark detector extracts watermark but it is "Never Copy"mark. So, the Media stream can not be copied.

Figure 5-18 shows that the watermark detector extracts watermark. The watermark is "CanCopy" mark. So, the Media stream can be copied.

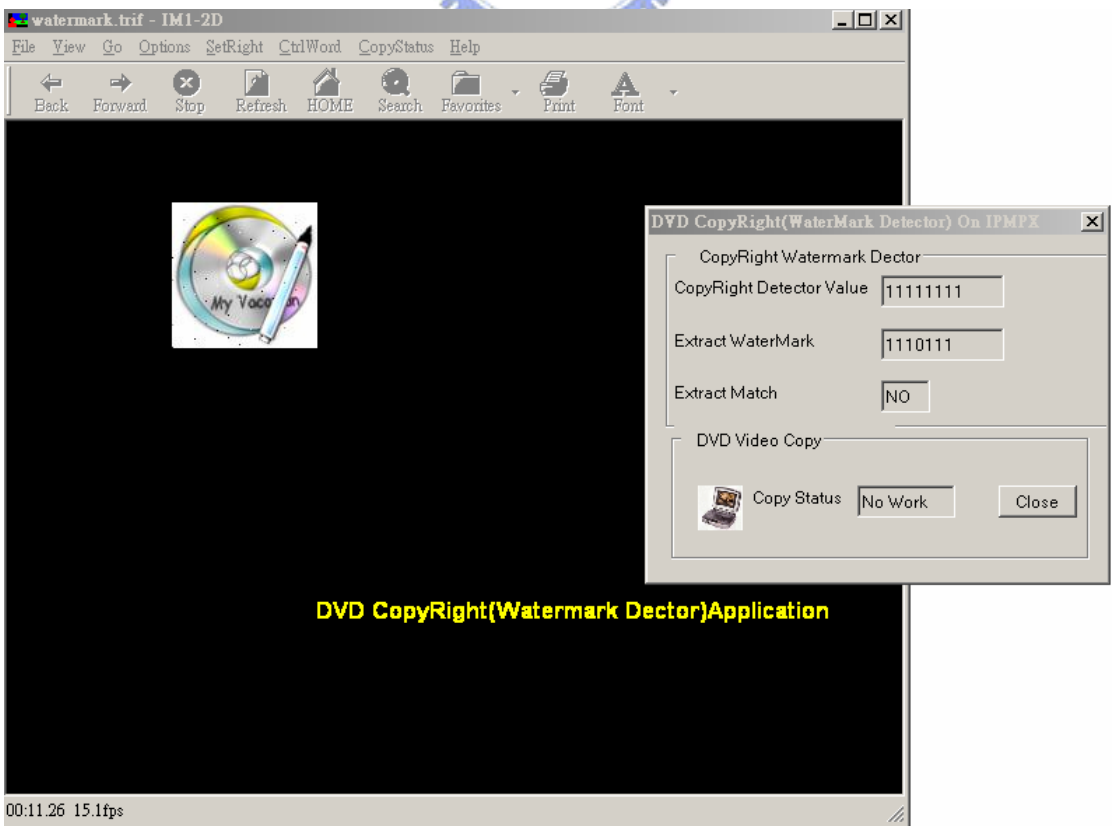


Figure 5-17 Demo 1 for DVD Copy Protection

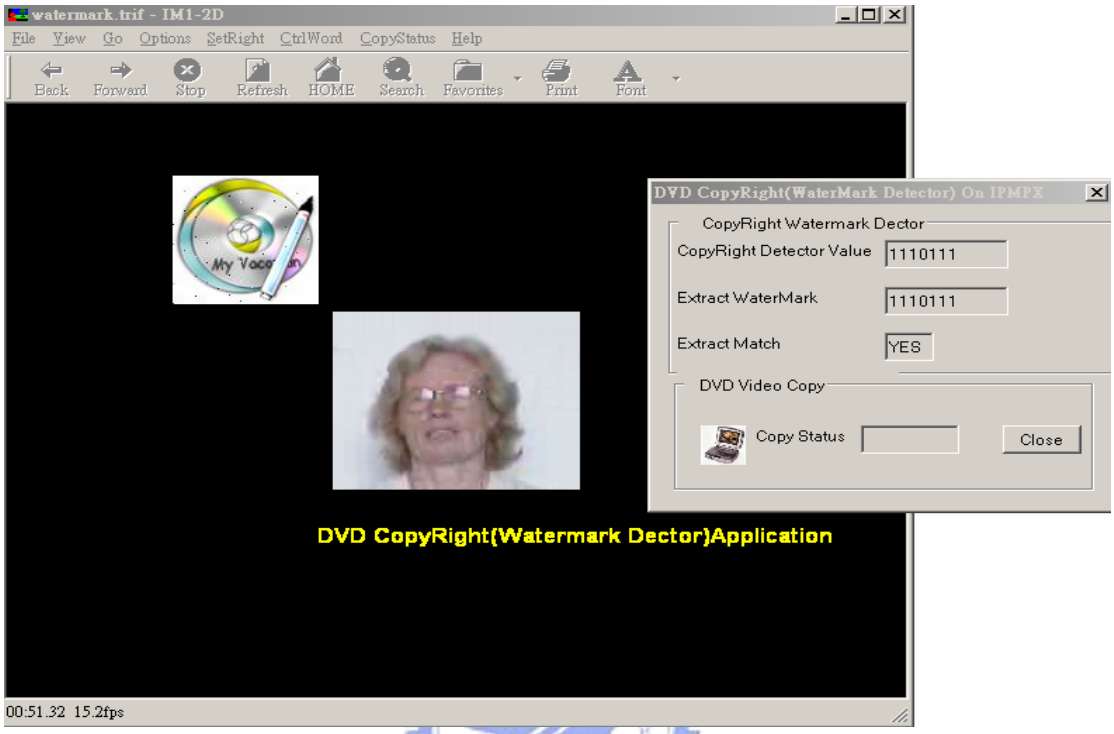


Figure 5-18 Demo 2 for DVD Copy Protection

5.3 MPEG-21 Rights Expression Language(REL) in an MPEG-4 IPMP-X

IPMP-X provides place holders for any rights language. As such, MPEG-21 REL can fit in. The current reference software demonstrates how an MPEG-4 IPMP-X terminal can use a set of IPMP tools to enable REL-License based rights management.

5.3.1 Introduction to MPEG-21 Right Expression Language

The Rights Expression Language (REL) has been an ISO standard on MPEG-21. It defines a machine readable, XML-based language for expressing rights. MPEG started the development process in 2001 with a Call for Proposals and an evaluation process in which XrML 2.0 was selected as the core architecture and starting point for the MPEG REL. It provide an authorization model to determine if a principal has the right to perform an action on a resource on condition according to REL Expressions(e.g. Bob has the right to play a video file for a week if he pays NT$100).It supports many business models in the end-to-end distribution value chain(e.g pay per view, library loan,…).

The MPEG REL is an important technology for the development of interoperability across DRM systems as well as Content Management and Digital Asset Management systems. With the MPEG REL one can express a wide variety of business models.

The next paragraph, we provide brief overview of the MPEG-21 REL data model and the structure of the language.

(1)REL Data Model

Using the MPEG-21 REL, anyone owning or distributing digital resources can identify principals(such as users, groups, devices, and systems) allowed to use those resources, the rights available

to those principals, and the terms and conditions under which those rights may be exercised.

The figure 5-19 describes the seven basic MPEG-21 REL elements and their inter-relationships.

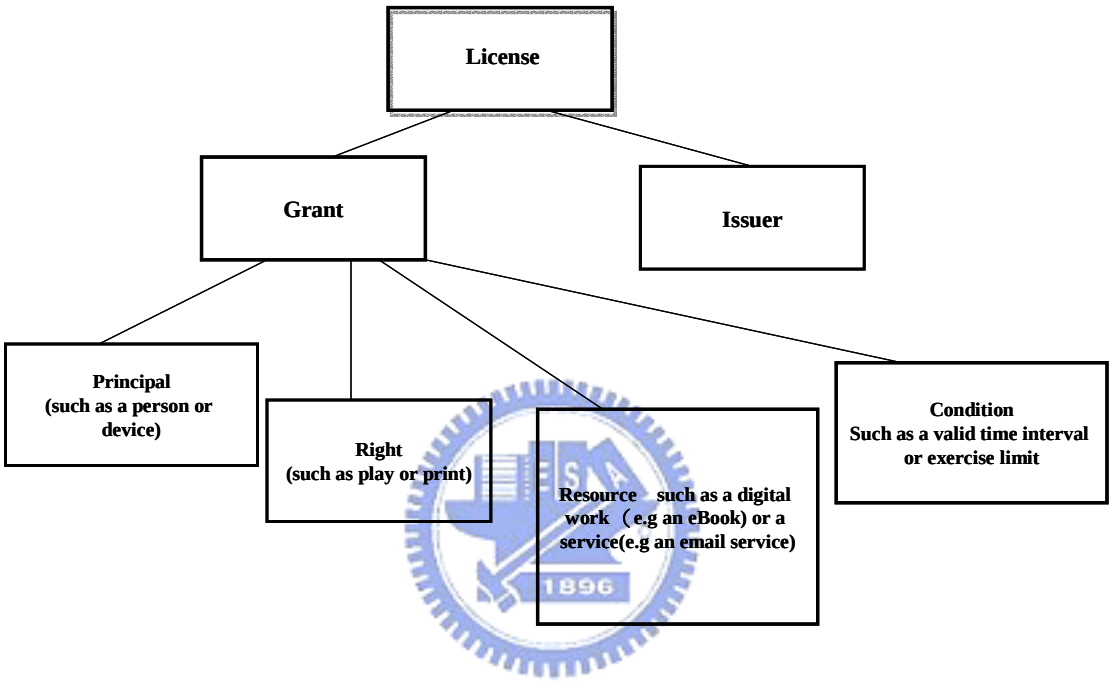


Figure 5-19 REL Data Model,[9]

A simple REL license looks like the scheme 5I in its skeleton form

Scheme 5I:

```
license
    grant
        Alice
        play
        aSong.mp3
        for 3 weeks
    issuer
        PDQ Records
```

Figure 5-20 The Example of REL License,[9]

In this example, a principal (Alice) has been granted the right to play a resource (a song) under the condition that she can only play it for 3 weeks. This right is

conveyed under the authority of the license issuer (PDQ Records). The license shown above provides only a simple illustration of the MPEG REL data model. It makes no attempt to fully illustrate the flexibility and expressiveness of the language. In fact, the MPEG-21 REL can be used to create licenses that address a wide variety of business models. Many web sites support example about REL (e.g http://www.contentguard.com, http://xml.coverpages.org/mpegRights.html).

(2) MPEG-21 REL Structure

The MPEG-21 REL is designed to be extensible and is itself specified in extensions. Its syntax is described and defined using the XML Schema and Namespace Recommendations by W3C [16],which enables the MPEG-21 REL to offer a high degree of richness and flexibility in its expressiveness and extensibility. A principal MPEG-21 REL design goal is to enable and support significant extensibility from the basic data model. The MPEG-21 REL is organized into several architectural parts as demonstrated in figure 5-21

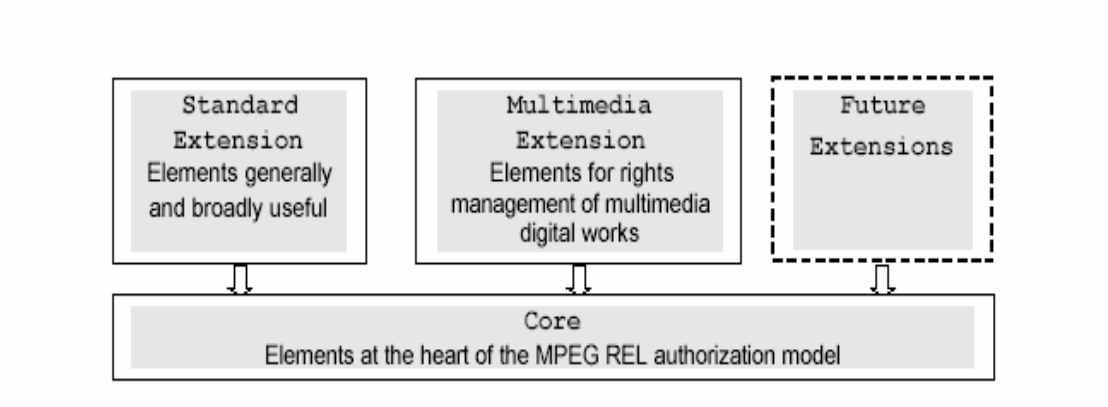


Figure 5-21 REL Extension Mechanism, [9]

The core, standard extension, and multimedia extension, as well as their XML Schemas, are normative parts of the overall MPEG-21 REL specification. Other

parties may, if they wish, define their own (possibly domain-specific) extensions to the MPEG-21 REL and its future extensions, as shown in figure 5-21. This is accomplished using the existing, standard XML Schema and XML Namespace mechanisms.

Core Schema defines structural and validation semantics that comprise the essence of the specification. It includes license, grant, principal , right ,resource ,condition and defines elements containing principal, right ,resource and condition in an abstract fashion.

Standard extension extends the condition type in the XrML core.It supports additional terms with the notion of external services required to exercise a right, payment conditions and methods, and time conditions (eg validityTimePeriod, paymentPeruse….).

Multimedia extension expands the Core Schema by specifying terms that relate to digital works. Specifically, the multimedia extension expands the rights, resources and condition.

(1) Rights:

1.render rights (e.g export ,play ,print,…)

2.transports right (e.g copy , loan ,transfer)

3.derivative work right(e.g edit ,embed ,extract)

4.file anagement(e.g backup ,delete,write ,execute,…)

5.configuration rights (e.g install ,uninstall)

(2) Resources:define terms for digital works and the metadata that describe the digital works

1. digitaWork
2. simpleDigitalWork Metadata

3. security Level

(3) Conditions: the exercise of rights for digital work (e.g destination, source, helper, render,…)

5.3.2 IPMP Master Tool and IPMP REL Tool for the MPEG-4 IPMP-X Framework[19]

The following example is provided by IM1 software reference. We extract the important code to explain how to work for MPEG-21 REL in MPEG-4 IPMP-X framework, but we do not really implement it because the code is not complete.

The application contains Master tool and REL tool.The Master tool is responsible for the REL license pathname extracted from the IPMP-X_Data field, the corresponding file is opened and the license is read and stored in a string. The major concept of this paragraph is in figure 5-22.

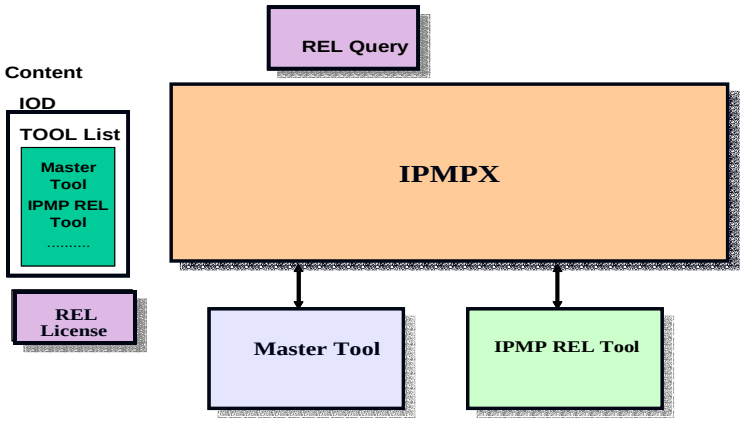


Figure 5-22 IPMP Master Tool and IPMP REL Tool for the MPEG-4 IPMP-X Framework

The REL tool notifies the Master tool. When the Master tool receives an IPMP_NotifyToolEvent message from the REL tool, then it set “CONNECTED” message to the REL tool. The Master tool sends first the license, and then the query

message . The application uses the method SendSecureMessage of an object of class SecurityLib though the encryption of the data. The Master Tool uses a separate thread for composing the IPMP_RightsData messages and for sending them. The major concept of this paragraph is in figure 5-22.

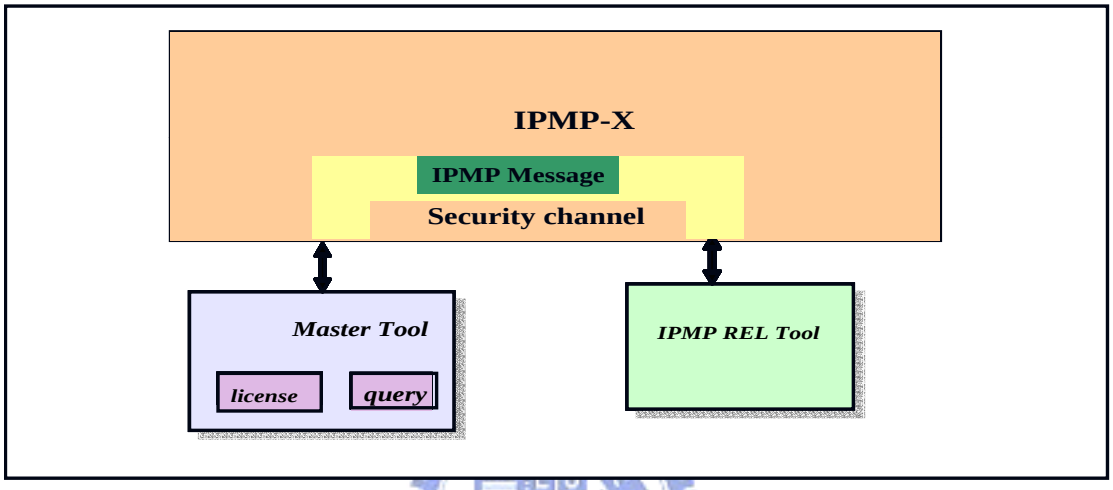


Figure 5-23 REL license and query contained in IPMP_RightsData message and transferred through security channel

The IPMP_RELTool is based on Microsoft MSXML4 parser.The current version supports only one Right “play” and one Condition “validityInterval”. The REL license and query are sent to the IPMP_RELTool inside IPMP_RightsData messages which are contained in IPMP_SecureContainer messages, but the mutual authentication and encryption/decryption tasks provided by the SecurityLib class are not implemented yet. Once the License and query are received and copied internally, the IPMP_RELTool performs the validation procedure. The flow control of validation procedure is described in Figure 5-24, Figure 5-17 and Figure 5-28.The result is sent back to the sender via IPMP_CanProcess message. The condition is described in Figure 5-25. Moreover, the IPMP_RELTool sends back to the sender the information extracted from the License and query during the validation process, using an

IPMP_OpaqueData message as shown in figure 5-26,once this message contains at least on matching grant was found.

The Master Tool will receive from the REL Tool an IPMP_SecureContainer message containing an IPMP_OpaqueData message conveying information extracted from the license during the validation process. In case no condition was met, the IPMP_OpaqueData will not be protected.

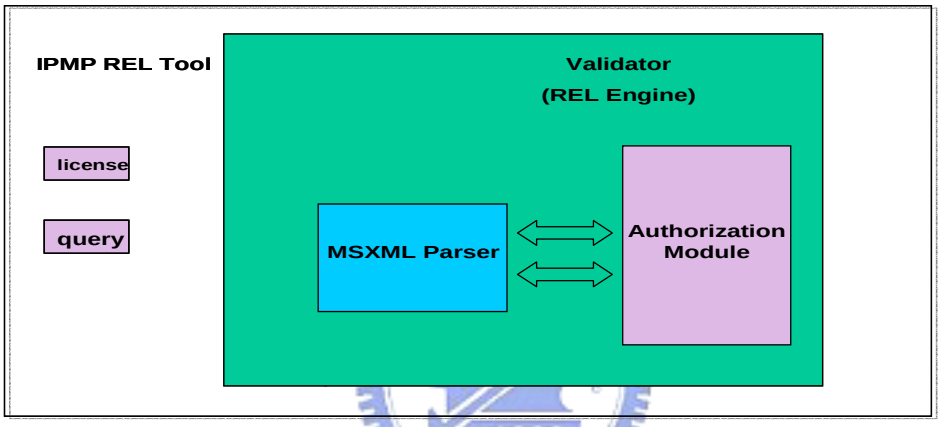


Figure 5-24 Extract items, validates request and conditions

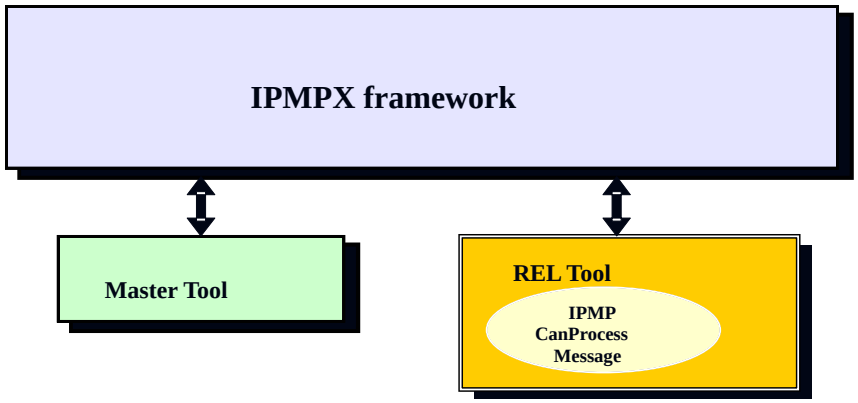


Figure 5-25 IPMP_RELTool sends back to Master Tool via IPMP_CanProcess

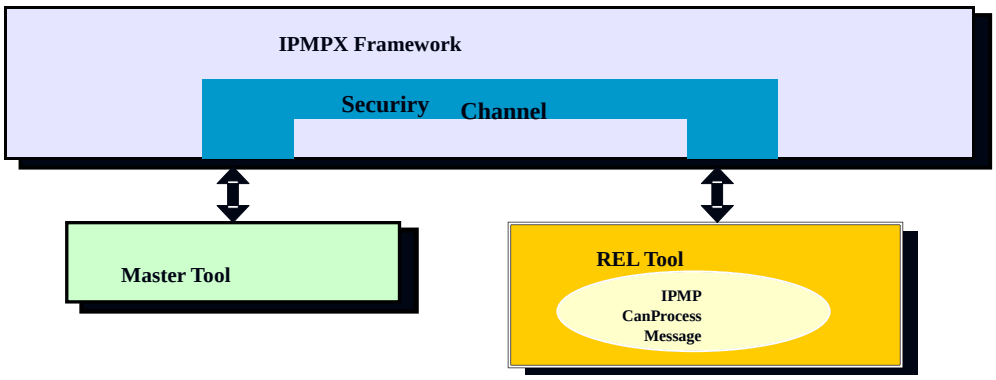


Figure 5-26 IPMP_RELTool sends back to Master Tool via IPMP_OpaqueData message

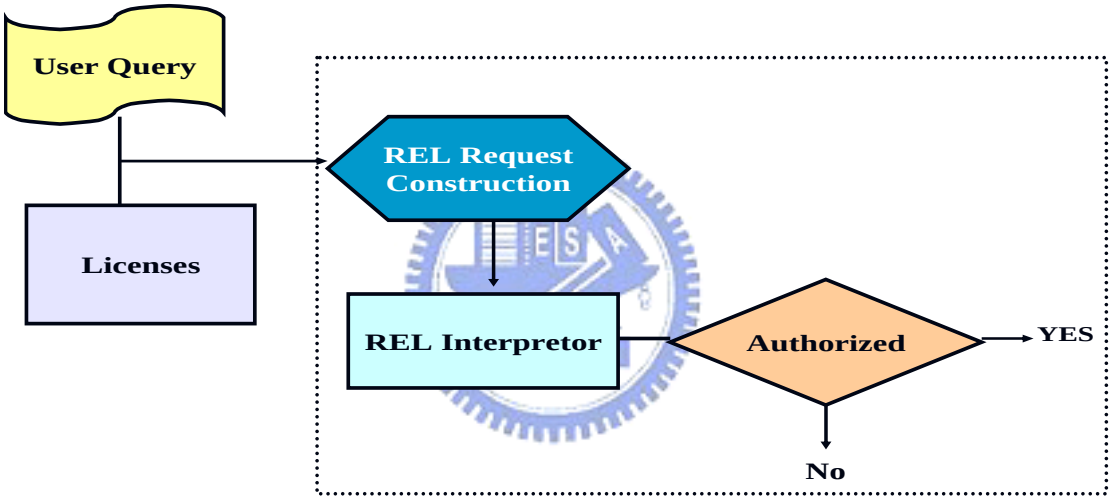


Figure 5-27 The control flow of REL tool in authorization

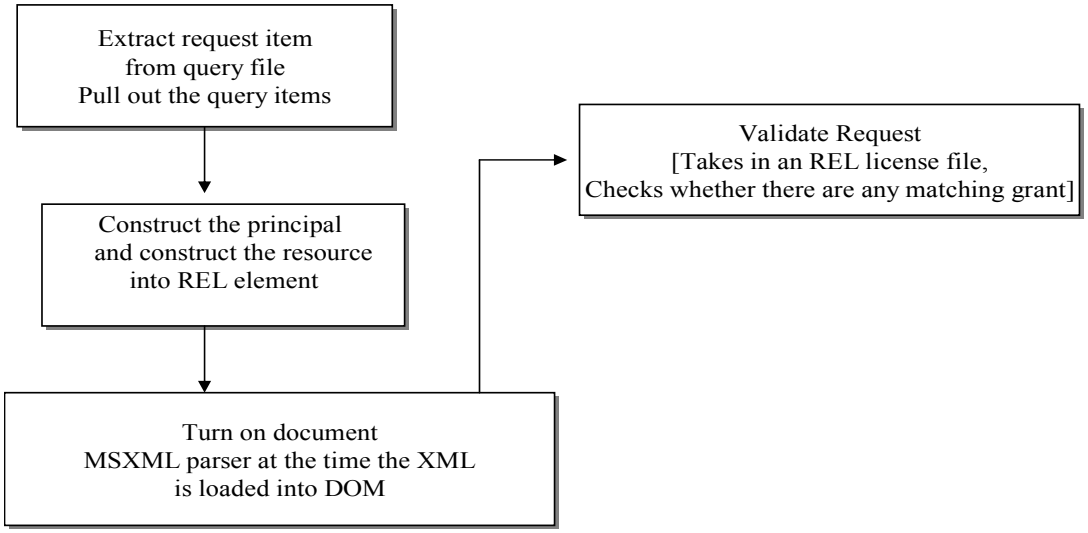


Figure 5-28 The control flow of REL tool for REL engine