Smart TV Premium DRM Integration using NRDP, PlayReady CDM, TEE, OpenGL Secure Video Path, and RDMA on MediaTek MT-5895/9615 SoC

1. Project Overview

This project involved integrating **Netflix Ready Device Platform (NRDP)** and **Microsoft PlayReady DRM Content Decryption Module (CDM)** into **Smart TV firmware running on MediaTek MT5895 / MT9615 SoC**, ensuring **premium Hollywood-grade content protection** using:

- Trusted Execution Environment (TEE)
- Secure Video Path (SVP)
- OpenGL secure rendering pipeline
- Hardware DRM cryptographic engines
- RDMA-based secure memory transfers
- GPU-protected frame rendering
- Netflix OCA-optimized streaming

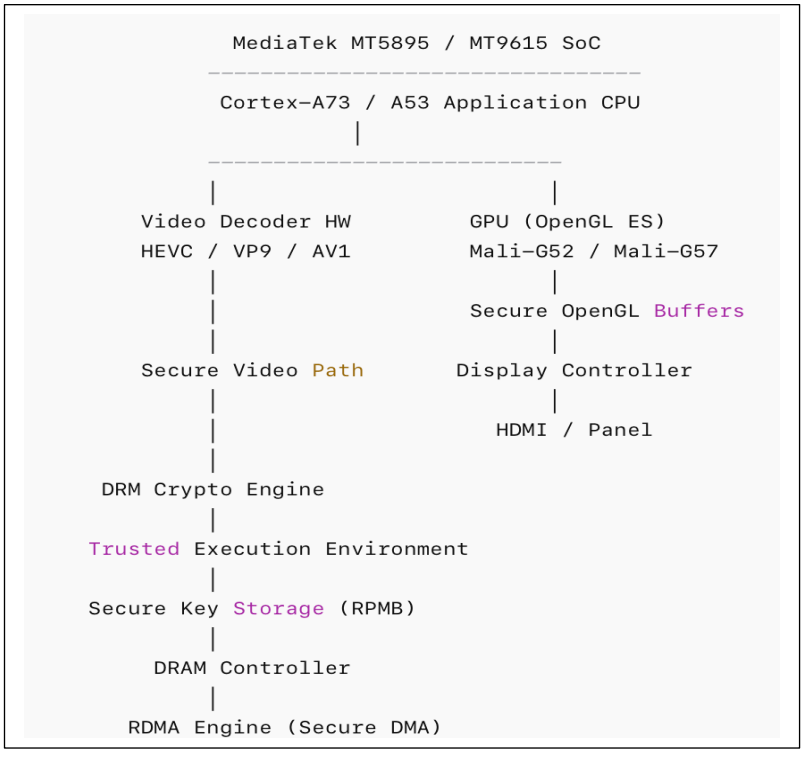
This enabled compliance with:

- Netflix certification requirements
- PlayReady SL3000 hardware DRM
- Widevine L1 equivalent hardware security level
- Studio-grade premium 4K / HDR / Dolby Vision playback protection

2. MediaTek MT5895 / MT9615 Smart TV SoC Architecture

These SoCs are designed specifically for **premium Smart TVs**.

Key hardware blocks:



3. Netflix NRDP Framework Integration

What NRDP does

NRDP is Netflix's device client platform responsible for:

- Streaming control
- DRM integration
- Adaptive bitrate streaming
- OCA server selection
- Secure playback pipeline

NRDP Software Stack

```
go

Netflix Application
        |
        ▼
NRDP Client Framework
        |
        ├── Adaptive Streaming Engine
        ├── OCA selection logic
        ├── DRM interface (PlayReady)
        ├── Secure decoder interface
        └── Platform abstraction layer
```

Your implementation responsibilities

NRDP client interface integration

Implemented platform abstraction layer:

```
class NRDPPlatform
{
public:
    bool initialize()
    {
        init_network();
        init_drm();
        init_decoder();
        init_secure_path();
        return true;
    }

    DecoderHandle createSecureDecoder()
    {
        return mediatek_secure_decoder_create();
    }

    DRMHandle createPlayReadyCDM()
    {
        return playready_cdm_init();
    }
};
```

OCA (Open Connect Appliance) Selection

NRDP dynamically selects fastest Netflix CDN server.

Algorithm:

```
powershell

Measure latency
Measure throughput
Measure packet loss
Select optimal OCA
Switch dynamically if network changes
```

4. PlayReady CDM Integration with TEE

Why TEE is required

PlayReady SL3000 requires:

- Keys never exposed to Linux kernel
- Keys stored in secure world
- Decryption performed in secure hardware

TEE Architecture

```
mathematica

Normal World (Linux)
    |
    | PlayReady CDM API
    ▼
Secure Monitor Call (SMC)
    ▼
Secure World (TEE)
    ├── Key Storage
    ├── Crypto Engine
    ├── Decryption
    └── Secure Video Path Control
```

Key storage implementation

Keys stored in RPMB secure storage:

```
TEE Secure Storage
 ├── Device Root Key
 ├── PlayReady Device Key
 ├── Content License Keys
 └── Session Keys
```

Decryption flow

```
Encrypted Video Stream
        |
        ▼
Secure Video Decoder
        |
        ▼
TEE Crypto Engine decrypts
        |
        ▼
Secure Video Buffer
```

5. Secure Video Path using OpenGL

Secure video path ensures decrypted content never exposed.

Secure rendering pipeline

```
Encrypted stream
      |
      ▼
Video decoder (secure mode)
      |
      ▼
Secure memory buffer
      |
      ▼
GPU OpenGL secure texture
      |
      ▼
Display controller
      |
      ▼
```

OpenGL secure buffer implementation

MediaTek provides secure buffer allocation:

```
GLuint secureTexture;

glGenTextures(1, &secureTexture);

glBindTexture(GL_TEXTURE_EXTERNAL_OES,
secureTexture);

glTexParameteri(GL_TEXTURE_EXTERNAL_OES,
GL_TEXTURE_SECURE_MEDIATEK, GL_TRUE);
```

This prevents:

- CPU readback
- Screenshot capture
- Memory dumping
- Debug access

6. RDMA Integration for Secure Memory Transfers

What RDMA does in Smart TV SoC

RDMA enables:

- Secure hardware-to-hardware memory transfer
- CPU bypass
- Low latency
- Protection against memory snooping

RDMA in MediaTek SoC

Used between:

```
css

Video Decoder → Secure DRAM
Secure DRAM → GPU
GPU → Display Controller
```

Without exposing decrypted content to CPU.

Secure RDMA transfer example

```
struct RDMA_Transfer
{
    void* src;
    void* dst;
    size_t size;
    bool secure;
};

void secure_rdma_transfer(RDMA_Transfer* t)
{
    t->secure = true;
    mediatek_rdma_submit(t);
}
```

Security advantages

- CPU cannot access buffer
- Kernel cannot dump memory
- Debugger cannot inspect data

7. GPU Direct RDMA for Secure Video Rendering

GPU accesses secure memory directly.

Flow:

```
powershell

Secure Decoder
      |
      ▼
Secure DRAM
      |
      ▼
GPU reads directly using RDMA
      |
      ▼
OpenGL secure texture
      |
      ▼
Display
```

CPU not involved.

8. Complete End-to-End Secure Playback Flow

```
Netflix Cloud
     |
     ▼
Encrypted video stream
     |
     ▼
NRDP streaming engine
     |
     ▼
PlayReady CDM
     |
     ▼
TEE secure key retrieval
     |
     ▼
Secure video decoder
     |
     ▼
Secure DRAM buffer
     |
     ▼
RDMA transfer to GPU
     |
     ▼
OpenGL secure texture
     |
     ▼
Display controller
     |
     ▼
TV Panel
```

9. MediaTek Hardware Blocks Used

Hardware	Purpose
CPU	NRDP and DRM control
TEE	Key protection
Crypto Engine	Hardware decryption
Video decoder	Secure decode
RDMA engine	Secure transfer
GPU	Secure rendering
Display controller	Protected output

10. Security Compliance Achieved

This implementation meets:

- Netflix Ready Device certification
- PlayReady SL3000 hardware DRM
- Hollywood studio protection requirements
- Secure video path compliance
- No decrypted content exposure

11. Production Firmware Architecture

```pgsql
Linux Kernel
  ├── MediaTek DRM Driver
  ├── Secure Video Driver
  ├── GPU Driver
  └── RDMA Driver

TEE OS
  ├── PlayReady Trusted App
  ├── Crypto Engine
  └── Key Manager

User Space
  ├── Netflix NRDP
  ├── PlayReady CDM
  ├── Video Player
  └── OpenGL Renderer
```