Tapwave® Digital Rights Management

Version 1.1a

1.

Overview

Tapwave anticipates that a significant portion of our customers will purchase and download commercial software from the Tapwave Store (an eCommerce service). Premier game content providers, publishers, and developers that are concerned about software piracy require a Digital Rights Management (DRM) system to be in place in order to offer commercial software for download.

Tapwave has created a DRM system that uses a software signing function to tie an application download to a specific hardware device. The signing process takes a generic copy of a licensed software title developed by a Tapwave partner and embeds a unique signature in the file before download. The signature ties the application to the target device's unique hardware serial number. This is the primary means of protecting licensed software.

There will be some software sold on an MMC memory card. These cards contain the software in an unchangeable form and they contain an unchangeable hardware ID. When the card is manufactured a unique signature is created and embedded into the application as it is placed on the card. This signature ties the application to the memory card's unique hardware ID.

Goals

- To enable secure distribution and sale of games and other application software for Tapwave products.
- To limit access to application software distributed by Tapwave to only those people who acquired a proper license to use it.
- To provide an easy, simple, and consistent user experience for secure distribution and use of software distributed by Tapwave.
- To maximize developer acceptance by being easy to integrate into an application.

Characteristics

The Tapwave DRM technology is based on industry standard and industrial strength Public Key Cryptography Standards (PKCS) from RSA Security. The security of the Tapwave signature generation system is ensured by strict firewall and physical access controls for the encryption engine and private key storage.

The Tapwave DRM approach is enabled by the presence, in each Tapwave device, of a unique, unchangeable serial number which is set during manufacturing. Access to this serial number is strictly controlled during operation to ensure that it cannot be spoofed by unauthorized software. Likewise the public keys used to validate Tapwave signatures are stored in a secure, unchangeable area within the Tapwave device that is protected from spoofing. Lastly, the key validation algorithms are contained in the Tapwave system software which is embedded in the device during manufacturing. During operation the system software periodically monitors itself to detect any unauthorized software that attempts to spoof a system function.

The Tapwave DRM technology provides three capabilities:

- **Tamper Resistance** - Ensures the integrity of a Palm OS application after its release.
- **Capability Access** - Controls application access to Tapwave protected capabilities.
- **Copy Protection** - Limits application execution to a specific hardware device.

In addition, the Security APIs available to Tapwave application developers enable hardening techniques that can foil sophisticated hacking attempts.

How does it work? – User's perspective

The online Tapwave Store includes:

- Browsing of available solutions or software packages
- Selection & purchase of software and payment handling
- Download delivery of the software that contains a signature specific to the customer's device.

The signature is automatically generated by the DRM system and is based on the unique hardware ID contained in the customer's device. This unique number is automatically discovered and recorded within the Tapwave eCommerce system when the customer first registers the Tapwave device. Subsequently, during the purchase and download of a software product from the Tapwave store the customer doesn't do anything to generate or embed the signature in the application. It is an invisible part of the download process.

When the software installation is complete, the device OS automatically verifies that the embedded signature matches the hardware ID. Again the customer doesn't do anything special. It is an invisible part of the operation of the Tapwave device.

2. Implementing DRM

There are multiple steps that a developer can pursue to take advantage of the Tapwave DRM technology. These steps range from simply adding a signature to adding sophisticated code. Below is an outline of the different levels you can support depending on your needs:

Level 1	Obtaining access to protected Tapwave APIs. Tapwave requires an application be signed to access the TwGfx and TwInput APIs. At this level, a developer is not concerned with copy protection or other security precautions.
Level 2	Supporting a reliable, simple to implement, and simple to use copy protection mechanism.
Level 3	Hardening an application to resist a sophisticated attempt at cracking the application and reassembling it without copy protection.
Level 4	Ensuring that the system software has not been corrupted or spoofed in order to defeat its copy protection mechanism.

The first and second levels add access and copy protection functionality to an application. This involves adding an application signature and a hardware signature. These levels are generally required if an application uses the Tapwave protected APIs or if you wish to sell and distribute your application through the Tapwave online store.

The third and fourth levels harden an application by adding code that uses security functions to perform additional verification of the integrity of the application and of the operating system.

More details are available in the "Digital Right's Management API" section of the *Tapwave Programmer's Reference* and in a HardenedSecurity sample application.

Level 1 - Signing an Application / Obtaining Access

Signing an application is the first step towards using the Tapwave DRM technology. Signing an application is necessary if your application accesses Tapwave protected APIs -- the TwGfx and TwInput API. A signed application is necessary to begin the Tuned for Tapwave compliance testing and subsequently to be available for sale via ESD at the Tapwave online Store. Signing an application is also the first step towards making your application resistant to corruption.

To sign an application you must first apply to obtain access to the Tapwave Signature Server. Please apply at http://www.tapwave.com/developers or contact Joyce Morrell (joyce@tapwave.com) for more information.

To make signing an application less cumbersome during the development process, you can download a Developer Access Application (DAA) via the Signature Server. The DAA unlocks a specific piece of hardware so that it no longer restricts access to the protected APIs. See **Error! Reference source not found.** below for more information.

Creating a Signed Application

Signing an application is very easy; you just submit your application to the Tapwave Signing Server (http://www.tapwave.com/developers).

There is a special resource that you may want to include before signing your application. This resource informs the Signing Server about how to sign your application.

> 'TSIG' #0 – Signature Skip List
>
> The Skip List resource identifies application resources that must be excluded from the signature generation -- this must include any resources that are modified during application execution. An example is an application that modifies its data to save a registration code provided by the user.
>
> If the Skip List resource is not present when the Application Signature is generated, then all the resources in the application are used to generate the signature. Note that you cannot include 'TSIG' resources in the skip list. Also note that if the most significant bit (MSB) of a resource type is set to 1, then it is automatically treated as if it is in the skip list.

The Skip List is a collection of one or more 8 byte entries. Each entry is of the form:

Resource Type	4 bytes
Resource Number	2 bytes
Pad	2 bytes – set to zero

The TSIG 0 resource can be created by Constructor for Palm OS (version 1.9). Create a new Custom Data resource and set the type to TSIG, the resource ID to 0 (zero), and the data source to Use Inline Data. Then fill in the binary data for the entries following the format above.

The Tapwave OS validates an application's signature from time to time during execution, including when the application makes calls to the protected APIs. This ensures that the application has appropriate access to these APIs and that the application has not been modified since the signature contained in the application was created. If the signature is invalid, the device resets and displays an error message.

At a minimum, we recommend calling one of the Tapwave Security functions (e.g.: `TwSecVerifyCurrentApp`, `TwSecVerifyDatabase`) to validate your application's signature. This function prevents a hacker from removing the signature resources from your PRC.

Developing with a DAA

The Developer Access Application (DAA) is a special tool that allows you to access the protected APIs without signing your application. Normally access to the protected APIs is allowed only to applications that have a valid application signature. During your development cycle it would be cumbersome to require generation of a new application signature for each iteration of the application. Presence of a valid DAA on your device enables you to bypass validation of the application signature.

Acquiring a DAA

You can acquire a DAA from the Tapwave Signing Server. The DAA contains a development signature that is created from your device's serial number and some developer information. Note that the DAA is locked to a specific hardware device.

Using a DAA

The DAA is a Palm OS application that you must install on your development device. When you launch the DAA on the device it validates its signature and displays the hardware ID of the device. The DAA does not need to be launched to perform its primary use. Its presence on the device is sufficient to bypass the OS verification checks.

The DAA contains a box labeled "Access Enabled." When this box is checked, the OS bypasses the normal verification procedure. Un-checking the "Access Enabled" box allows a developer

to check whether the OS is correctly validating the application, whether the application is using a valid signature, and whether the application is using a protected API.

Level 2 - Locking an Application/Copy Protection

Locking an application is a follow on step in using the Tapwave DRM technology (a locked application must also be a signed application). A locked application contains a hardware signature that locks execution of the application to a specific Tapwave device. A hardware signature can also lock execution of the application to the presence of a specific ROM card.

Creating a Locked Application

The hardware signature is generated when a user downloads an application from the Tapwave store via an interaction with the Tapwave Signature Server. It is also generated during ROM card production.

There is a special resource that you must include in your application to generate a hardware signature.

'TSIG' #1 – Hardware Signature Required

The presence of this resource instructs the OS to validate that an application is running on a specific hardware device.

If the hardware signature is not present, or if its validation fails, the system resets and reports the error to the user. This prevents the application from successfully running on any device other than the ID matching the hardware signature. This mechanism is also used for applications that are delivered on ROM cards so that the application is tightly tied to a specific card.

The Hardware Signature Required is of the form:

Version	1 byte – set to 1
Type	1 byte
LockingRequired	1 byte

Where Type is:

0	none allowed.
1	device signature required
2	card signature required
3	allow device or card locking
FF	allow any locking type

And LockingRequired is:

0	locking is not required
1	locking is required

Setting the LockingRequired byte to 1 means that the device resets if the application is running on a device to which it is not locked. However, if you set this byte to 0, the device does not reset but TwVerifyDatabase returns `twResetReasonInvalidOptHwrSig`. This gives you an opportunity to run your application in a degraded demo mode if it has been copied to another device.

Note that if no hardware signature is found, the OS behaves as if the LockingRequired byte is set to 1.

The TSIG 1 resource can be created by Constructor for Palm OS (version 1.9). Create a new Custom Data resource and set the type to TSIG, the resource ID to 1 (one), and the data source to Use Inline Data. Then fill in the binary data following the format above.

The Tapwave OS validates an application's hardware signature from time to time during execution, including when the application is launched. This ensures that the hardware signature contained in the application is consistent with the hardware ID of the device on which it is executing. Thus if a locked application is copied to another device it does not execute normally. It may simply exit or it may run in some degraded mode depending on the developer's choice.

Developers may choose to take a more proactive approach to copy protection and allow an application to run in some limited or demo mode when it is not executing on the device to which it is locked.

This locking mechanism is meant to replace any other locking or copy protection mechanisms used by your application. Having multiple copy protection mechanisms creates a confusing user experience. When running on a Tapwave device, your application should disable other copy protection mechanisms -- the HardenedSecurity example shows how to determine whether your application is running on a Tapwave device.

Level 3 - Hardening an Application

Hardening an application is the next step an application can take in using the Tapwave DRM technology (a hardened application must also be a signed application; it could also be a locked application).

A hardened application uses custom code and signatures to provide further assurance that the application cannot be disassembled, stripped of DRM, and then reassembled as a non-protected application.

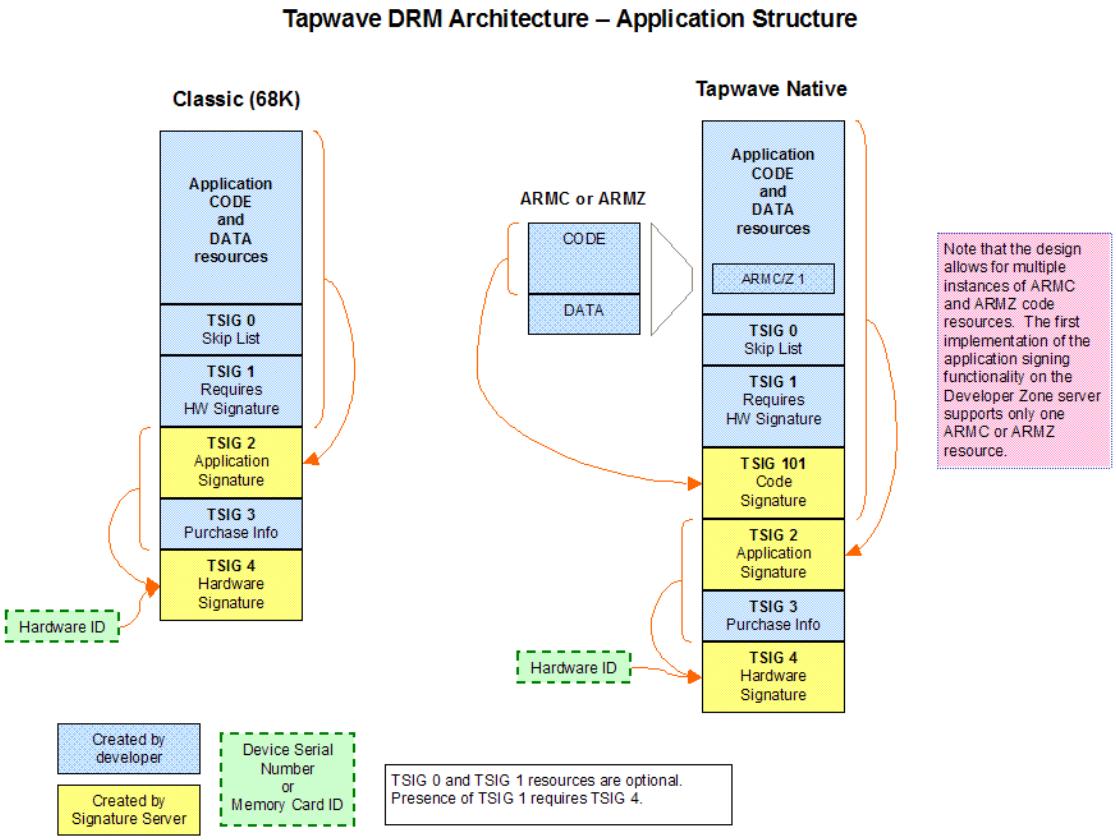
Verifying Application Integrity

The developer can use a variety of techniques to protect the PRC file. For example, create custom signatures from one or more data or code resources within the application, and then include these signatures somewhere within the application. When the application runs on a device, custom code within the application uses the Tapwave security APIs to validate special

application specific signatures that the developer created. These validations are performed at various places and times within the application code to radically increase the difficulty for another developer to discover and disable each of them. The HardenedSecurity example shows how to implement a variety of techniques.

Level 4 - Validating the OS

Validating the OS is the last step an application can take in using the Tapwave DRM technology. The application may use security mechanisms to ensure that the system software is operating as expected. The HardenedSecurity example shows how to implement a variety of techniques.



3.DRM Glossary

Application Signature
A small piece of encrypted data created only by the Tapwave Signing Server from a copy of the application executable file and a secured, private encryption key. The public companion to the private encryption key is stored in the ROM of each Tapwave Zodiac device.

Development Access Application
A Palm OS application (DevAccess.prc) that has a development signature from the Tapwave Signing Server incorporated appropriately into the executable file. Presence of this application allows any application on the device to access protected APIs that are present in Tapwave devices. This allows a developer to build and test iterations of an application without each one being a Signed Application.

Development Signature
A small piece of encrypted data created only by the Tapwave Signing Server from a developer information resource (Device Serial Number, developer information and a secured, private encryption key). The public companion to the private encryption key is stored in the ROM of each Tapwave device.

Device Serial Number
A unique and unchangeable number burned into each Tapwave device during the manufacturing process.

DRM (Digital Rights Management)
In the Tapwave development environment, DRM is a collection of technologies and processes that protect digital content and enables the secure distribution and sale of this content on the Internet.

The Tapwave DRM technology provides three capabilities:

Tamper Resistance - Ensures the integrity of a PalmOS application after its release

Capability Access - Controls application access to Tapwave protected capabilities

Copy Protection - Limits application execution to a specific hardware device

The first two of these are provided together via a signed application. The third is provided via a locked application.

ESD (Electronic Software Distribution)
The process of purchasing and distributing software from an online store such as store.tapwave.com. ESD through the Tapwave store utilizes DRM technology to download a Locked Application to the customer.

Hardware ID
This is either a device serial number or a memory card ID.

Hardware Signature
A small piece of encrypted data created only by the Tapwave Signing Server from a copy of the application signature, a hardware ID, and a secured, private encryption key. The public companion to the private encryption key is stored in the ROM of each Tapwave device.

Locked Application
A Palm OS application that has a hardware signature from the Tapwave Signing Server incorporated appropriately into the executable file. A Locked application must also be a signed application. A valid hardware signature ensures the application executes only when a specific, associated hardware ID is present. Thus the hardware signature locks execution of the application to a specific device. Alternatively, a hardware signature locks execution of the application to the presence of a specific ROM card.

Memory Card ID
MMC ROM and SD cards can be created with an unchangeable ID number. SD cards have unique ID numbers for each card instance. MMC ROM cards have a unique ID number for each batch of cards that contains the same image of data, e.g. all the cards that contain the same copy of a game and its data. The ID number is protected from copying to a RAM card.

Signed Application
A Palm OS application that has an application signature incorporated appropriately into the executable file. Presence of a valid application signature allows the application to access protected APIs that are present in Tapwave devices. A Signed application is also resistant to tampering. After the application is signed any changes made to the application code or data are detected at runtime on a Tapwave device.

System Signature
A small piece of encrypted data created only by the Tapwave Signing Server from a copy of the production version of the system ROM and a secured, private encryption key. The public companion to the private encryption key is stored in the ROM of each Tapwave device.