Signing Process Guide

Connected Sentinel Player 3.x

viaccess·orca

copyright

trademarks

Document reference number: 21716

Document version number: 1.2 Release

Contents

Document Evolutions

This chapter contains information on the modifications and updates applied to the manual since version 1.0.

Version 1.2

The following table lists the modifications made to version 1.2 of the manual.

Description of modification	Section concerned
Changed the version of the Connected Sentinel Player	Throughout manual

Version 1.1

The following table lists the modifications made to version 1.1 of the manual.

Description of modification	Section concerned
Modified the definition of the Connected Sentinel Player	*Glossary* on page 8

Chapter 1: Introduction

This guide explains the process of signing the SO (shared object) files that are used in parallel to the CSP-SDK (Connected Sentinel Player SDK). This procedure is mandatory for using the CSP-SDK.

SDL is based on build-time signing of binary modules followed by load-time verification of these modules. By ensuring that only properly-signed modules are loaded and used by the application, the mechanism blocks classes of code-injection attacks that use the dynamic loading interfaces.

As an obvious extension, the secure dynamic loading mechanism also covers signing and verification of developer-controlled configuration files, to prevent those from enabling attacks (by malicious changing of security-relevant configuration parameters).

The secure dynamic loading mechanism integrates with a separate secure runtime code-monitoring mechanism, intended to detect post-loading attempts to modify the code in memory.

Target Audience

This document is aimed at developers writing a player application based on the Connected Sentinel Player SDK and need to add an SO file/s to their application.

Glossary

This manual contains a lot of acronyms or terms that are specific to the field of Viaccess-Orca. If they are not defined within the text, refer to the *Glossary* on page 8 at the end of the manual for a complete definition.

Chapter 2: Signing Process

Overview

The diagram below explains the Connected Sentinel Player signing process:

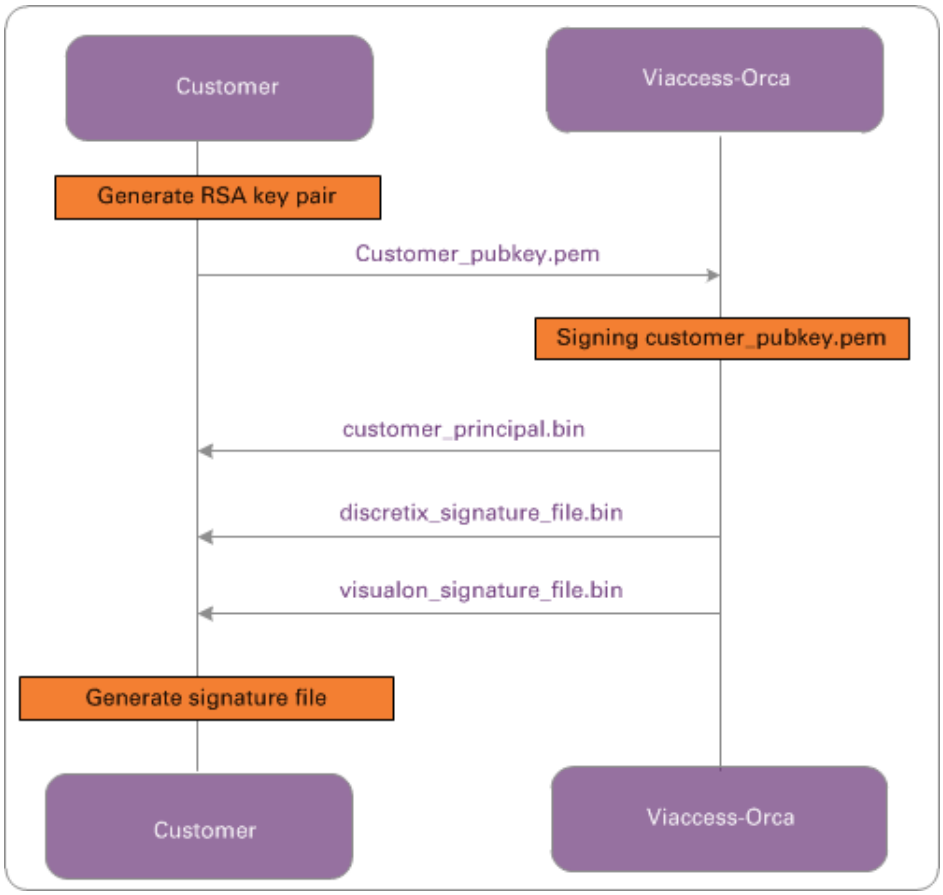

figure 1. CSP signing process

Generate RSA key pair

The RSA key pair should be generated once as follows:

note

<Name> should be replaced by something that represents your organization and MUST be consistent throughout the entire process.

note

<RSA password> is the password of your choice to protect the private key.

```
openssl req -newkey rsa:2048 -keyout <Name>_prikey.pem -passout pass:<choose password> -subj "/CN=<Name>" -out <Name>.pem
openssl req -in <Name>.pem -verify -noout -pubkey -out <Name>_pubkey.pem
```

The command operations yield <Name>.pem, <Name>_pubkey.pem, <Name>_privkey.pem

Only <Name>_pubkey.pem will be sent to Viaccess-Orca (this is the only key that should be sent to Viaccess-Orca as part of the SDL process).

Generate Signature binary

All SO files that the application needs to load must be added to the signature binary.

The libDxSig.so generated by the tool should be added to the project under libs/armeabi-v7a to enforce the authenticity of every library loaded to the runtime memory.

The signature binary is generated by `DxDlcSignatureFileGeneratorTool.exe` and should be generated as follows:

```
DxDlcSignatureFileGeneratorTool.exe -p <RSA password> -key <xxx_prikey.pem> -key-
sig <xxx_principal.bin> -v <SecurePlayer Package Name> -f <First SO path> -f
<Second SO path> -f <Third SO path> …… -sigf viaccessOrca_signature_file.bin -
sigf visualon_signature_file.bin -o libDxSig.so
```

note

The files <xxx_principal.bin>, `viaccessOrca_signature_file.bin` and `visualon_signature_file.bin` will be supplied by Viaccess-Orca after the customer sends to Viaccess-Orca the <Name>_pubkey.pem file.

note

<RSA password> is the same password used when generating the RSA key pair. This <RSA password> parameter can be omitted for the command line but the executable will demand the user to type the password manually during runtime.

note

Aside from the libDxSig.so, the executable generates a libDxSig.so.info file which is almost identical to the libDxSig.so with a text header of the parameters related to its creation. It is mainly used for diagnostics and should not be placed in the production binaries.

example 1

```
DxDlcSignatureFileGeneratorTool.exe -p 123456 -key C:\Dir\<Name>_prikey.pem -
keysig C:\Dir\<Name>_principal.bin -v GENERAL_ANDR_VOP_PROB_RC_02_00_00_0000 -f
C:\Dir\lib1.so -f C:\Dir\lib2.so -f C:\Dir\lib3.so -sigf
C:\Dir\discretix_signature_file.bin -sigf C:\Dir\visualon_signature_file.bin -o
C:\Out_Dir\libDxSig.so
```

example 2

```
DxDlcSignatureFileGeneratorTool.exe -p 123456 -key C:\Dir\<Name>_prikey.pem -
keysig C:\Dir\<Name>_principal.bin -v GENERAL_ANDR_VOP_PROB_RC_02_00_00_0000 -f
C:\Dir\lib*.so -sigf C:\Dir\viaccessOrca_signature_file.bin -sigf
C:\Dir\visualon_signature_file.bin  -o C:\Out_Dir\libDxSig.so
```

note

This last example shows the usage of wildcards in the -f parameter

Glossary

Definitions of technical terminology and acronyms are listed in the table below:

Term	Definition
Connected Sentinel	Viaccess-Orca's TV Everywhere Digital Rights Management (DRM) solution that supports both VO's proprietary DRM technology VO DRM and Microsoft® PlayReady®. With Connected Sentinel, content service providers can distribute content securely to STBs, tablets and smartphones running VO DRM, as well as to Microsoft-based terminals such as PC's and gaming consoles supporting PlayReady®.
CSP	Viaccess-Orca Connected Sentinel Player, i.e. the Discretix® (DX) Secure Player duly distributed by Viaccess-Orca.
SDL	Secure Dynamic Loading
SDP	Session Description Protocol
SO	Shared Object