***NextLabs, Inc.***

**Rights Management**

**Client Initial registration**

**Workflow**

# Change history

|  |  |  |
| --- | --- | --- |
| Date | Author | Description |
| 3/12/2015 | Philip Qi | * Adding deployment model * Updated registration flow after discussion |

Table of Contents

[Change history 2](#_Toc413937296)

[Deployment model 4](#_Toc413937297)

[Before registration 5](#_Toc413937298)

[Registration workflow 5](#_Toc413937299)

[Certificate chain 7](#_Toc413937300)

[Registration data 8](#_Toc413937301)

# Deployment model

In the first release, deployment model is administrator download a tool “Package Builder” from RMS and build a pre-configured RMC package via this tool. The generated RMC package should have all information on how to communicate to RMS to register itself and download policy for local policy evaluation.

In detail, administrator login to RMS and go to an RMS page to download a tool called “Package Builder”. RMS embedded RMC initial registration certificate in this tool when generating downloadable package.

Administrator then build RMC package by using the downloaded “Package Builder” tool.

The “Package Builder” tool generates standard MSI based package that can be used for any client management tool to deploy to end user’s desktop and laptop.

# Before registration

There are two ways to create Rights Management Client (RMC) install package. One is via a Windows application “Package Builder”, another is via Rights Management Server (RMS) web installation. In the first release, we only support using “Package Builder” tool to build RMC client.

Regardless how installation package is created, there are three important files related to RMC initial registration. They are:

* RMC initial registration certificate issued by RMS
* NextLabs root CA certificate (public certificate)
* RMS intermediate CA certificate (public certificate)

While RMC initial certificate private key and RMC initial certificate are created on demand when administrator download package builder tool from RMS, NextLabs root CA certificate and RMS intermediate CA certificate are pre-created and ship with the “Package Builder” tool and RMS.

“Package Builder” tool place RMC initial registration certificate when generating RMC package. RMC use this initial registration certificate to register to RMS and receive its own unique certificate for later communication.

# Registration workflow

RMC is going to register to RMS when RMC service first start after installation. RMC authenticates to RMS and receives its configuration, a unique RMC certificate and a unique RMC id for later communication between RMC and RMS.

The follow is the registration workflow.

|  |  |  |
| --- | --- | --- |
| **Step** | **Description** | **Workflow** |
| 1 | RMC connects to RMS via HTTPS.  RMC presents its initial registration certificate to RMS for RMS to verify.  RMS presents its certificate to RMC for RMC to verify. |  |
| 2 | RMC generates a session key.  RMC encrypts generated session key with RMS public key stored in received RMS certificate.  RMC add encrypted session key in base64 encoding in HTTPS request header.(X-NXL-S-KEY)  RMC encrypt registration information by using the generated session key and sends encrypted information as to RMS.  RMS gets session key from HTTPS header.  RMS decrypt session key by using its private key.  RMS decrypt received content by using decrypted session key. |  |
| 3 | RMS verifies decrypted content.  RMS finish registration process on its end.  RMS generates a unique GUID to uniquely indent this RMC.  RMS generates a unique RMC certificate and embedded it in the registration response XML.  RMS generates HTTPS response and encrypt it with the session key RMS received from RMC in step 3.  RMS sends response via HTTPS channel back to RMC to finish the registration process. |  |

# Certificate chain

There are three layers in rights management certificate chain. They are:

* Root certificate authority (Root CA)
* Intermediate certificate authority (Intermediate CA)
* Individual certificate

Root CA issues Intermediate CA. One Intermediate CA is assigned to a function unit such as “Package Builder” or RMS.

Intermediate CA issues individual certificate to RMC to represent RMC in later communications.



# Registration data

RMC sends encrypted information to RMS via HTTPS when it register to RMS. The format of the registration request is defined in “REST Web Service for ‘RegisterAgent’” spec.