netUpdate
Security Specification

Version 2.0

UpdateLogic Inc.
(508) 624-8688 (TEL)
(508) 624-8686 (FAX)
http://www.updatelogic.com

1 Confidentiality and Restricted Distribution

This document contains information that is confidential and proprietary to UpdateLogic Inc. (ULI). It is It is only made available to parties that have signed UpdateLogic, Inc.'s Non-Disclosure Agreement and may not be reproduced in any form without express written consent of ULI. No transfer or licensing of technology is implied by this document.

2 Change Tracking

Version Number of This Document	Changes
2.0	Original

3 Table Of Contents

4 Definitions

This section provides a glossary of terms used in this document.

Term	Definition
Agent	The UpdateLogic software that resides on a device and enables it to talk to the NOC and perform provisioning functions.
CEM	Consumer Electronics Manufacturer
CSP	Content Service Provider
CSP Client Program	A CSP specific program that manages handles session authentication and control and manages media access and display.
CV	Chip Vendor
Device	An Internet connected appliance that requires provisioning data.
Device Class	A class of the same devices. Same as a model. Device Class is scoped by company.
DRM	Digital Rights Management
NOC	Network Operating Center – a system of hardware and software that resides in the Internet cloud that a device contacts to receive its keys.
Provisioned Object	A device-unique or device class-common DRM-related object that CSP client programs rely on to authenticate sessions or encrypt media. Provisioned objects need to be handled securely due their use in DRM or e-commerce. Provisioned objects may be imported or generated by the NOC. Examples of provisioned objects include X.509 certs, X.509 keys, proprietary DRM related objects, or cryptographically random material.
Provisioned Object Class	A way to refer to all the provisioned objects of the same type across all devices in a given device class.
Provisioned Object Name	A zero-terminated ASCII string that identifies a provisioned object class.
Provisioned Object Owner	A zero-terminated ASCII string that identifies the owner of a provisioned object class.
Provisioning Request	A request made from a registered device to the NOC to receive device-unique data.
Query Host	A OUI and model group specific domain and subdomain that is purchased and maintained by

	ULI that connected devices use to contact the production NOC.
Registration	The process that all devices must go through when they first connect to the Internet in the consumer's home in order to register their identity with the NOC. Registration takes place regardless of whether a device is enabled for provisioning or not.
Supervising Program	The program in the device that controls the Agent.
SoC	System on Chip
ULI	UpdateLogic, Inc.
ULPK	UpdateLogic Provisioning Key. A device-unique key that is inserted at the factory to authenticate a device when it registers with the NOC.

5 Introduction

This document provides a description of netUpdate's security features.

6 The netUpdate Facility

netUpdate is the update component of the netReady suite of services. In case of a security breach in any part of the software on the device the netUpdate Agent can update the compromised component to close the security hole. NetUpdate takes advantage of three types of update delivery mechanisms: Internet, Broadcast, and USB. All three delivery mechanisms use the same tamper-proof, encrypted, security model.

6.1 General Information about Updates and Software Versions

Internet and broadcast updating occurs automatically and invisibly when the TV is powered off using the remote's power key. Due to concerns about simplifying the support matrix and security agreements with streaming media partners there is no way for a TV update to be rolled back. It's a one-way operation. If bugs are found in a recent update another update will be made to fix those bugs and the system will be updated again. When an update is complete the user will be notified that their TV software has been updated and the version string and a brief release note will be displayed. The current version string is usually displayed in a user menu. If a TV is left powered up then it will not receive updates.

6.2 Internet Updates

When the TV is powered down it checks for Internet updates first. The time an internet update takes to complete is dependent on the size of an update and the consumer's internet connection speed.

6.3 Broadcast Updates

If the NOC cannot be reached the best case time for delivering a broadcast update is 2.5 hours. The worse case time is 6.5 hours given the current load on the UpdateTV broadcast carousel. If the TV is power on during a broadcast update it saves the amount of data that it has received so far. When the TV is then powered down again the broadcast update picks up where it last left off. On TVs that are left powered up and down many times per day it may take a few days for the TV to receive an update. Updates do not affect the function of the TV until all of the parts of the update are received. Broadcast update channels are identified during the channel scan that takes place during onboarding. If a TV is moved to another city, the channel that the broadcast Internet updates occur automatically when the TV is shut off.

6.4 USB Updates

Updates can be delivered by USB. A consumer may download an update from a website and place it on the USB themselves or a USB stick can be mailed to them directly from the CEM. USB updates are unlike broadcast and Internet updates because they're initiated by plugging a USB stick into the TV when the TV is powered-up. The TV will check the USB stick for a new update and if it finds one it will tell the consumer about the version of the update it just found and will then power down automatically to receive the update. USB updates take around 45 seconds to complete although the total time will be a little more because the TV has to power down to get the update and then power back up to install it once it's been received.

6.5 Update Diagram

The diagram below outlines the update process.

6.6 Component Manifest

The component manifest is an encrypted, tamper-proof, ULI proprietary file that contains the following information about a devices's UpdateLogic identity:

- OUI
- Model Group
- NOC contact URL
- Update Step
- Software Version
- Component SHA256 hashes that lock the file to a particular update

The component manifest is created by the Publisher tool. A "native" component manifest is stored in the root file system of the device when it ships. Each update that occurs replaces the component manifest so that it properly reflects the attributes of that update. The component manifest is encrypted so that it can't be tampered with. The component manifest contains SHA256 hashes of all of the major components contained in a given update which locks it to a given software distribution. This lock mechanism prevents one component manifest from being replaced with another.

6.6.1 Rollback Prevention

Rollback prevention is an integral part of netUpdate. A running "update step" (also called a module version) increments with each update. The update step is kept by the NOC and stored in the component manifest. Even though a software version may be rolled back during an update the update step will always increment. The netUpdate Agent will not accept an update with an update step that is less than or equal to the update step stored in the active component manifest.

6.6.2 Minimum Software Version for Provisioning

The NOC will not deliver provisioned objects to a device that is running code that is less than a minimum software version. Therefore even if a compromised cramfs.img was copied into flash manually bypassing the UTV Agent checks all stored credentials would be revoked when the NOC found that an unauthorized version of the system was running.

6.7 Update Encryption and Authentication

NetUpdate's security features include the following:

- 2048 bit RSA encryption used to protect the payload decryption key of each module.
- AES128 or other HW-specific encryption is used to protect module payload data.
- SHA-256 secure hash algorithm to validate module payload data
- SHA-256 secure hash algorithm to validate each component which is made up of multiple modules.
- User name/Password-protected Publisher tool that authenticates a user via the NOC to encrypt the update modules using by OUI/Model Group
- Password-protected web site accessed through SSL to safely distribute encryption keys to Publisher.

6.8 Nested Encryption Ciphers

RSA asymmetrical encryption is inherently slow due to the nature of the mathematical operations performed using very large numbers. In addition to performance issues, the size of an encrypted block of data may be larger than the size of the original data (though that growth can be

controlled) requiring more bandwidth to transmit the encrypted data stream. Using a large enough key, RSA encryption is extremely secure, though the larger the key, the lengthier the encryption/decryption process. Symmetric encryption via a block cipher is much faster than RSA does not produce encrypted output that is larger than the original data. For these reasons, a hybrid method utilizing RSA encryption and AES128 encryption is utilized to secure updates.

Each data module is protected with RSA2048 encryption of its AES128 or other hw-specific symmetric keys. This key in turn is used to encrypt update payloads. This nesting allows the payload key to be changed with every update.

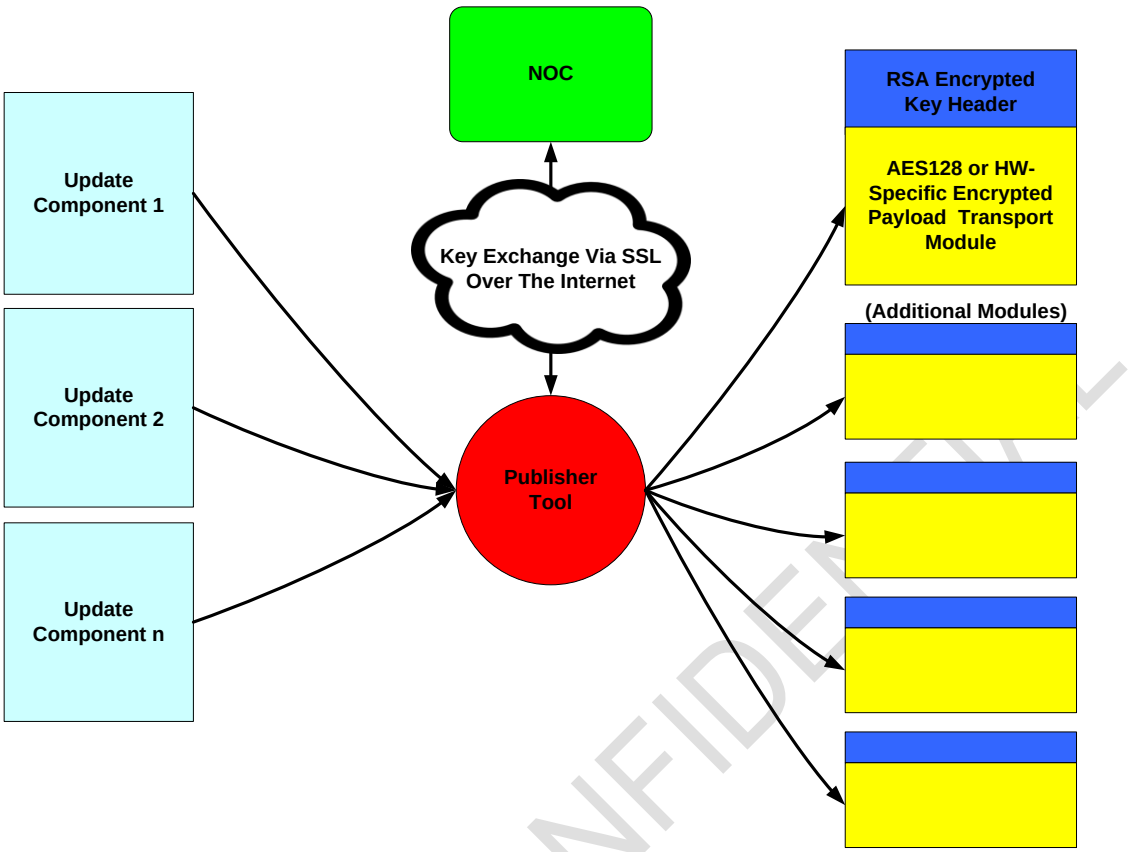
6.9 Secure Payload Key Access Via The NOC

The UpdateLogic NOC is responsible for generating and distributing the encryption/decryption keys used by netUpdate to secure update payloads. Each OUI/Model Group combination has its own set of RSA encryption/decryption keys. The NOC's web service provides a secure interface for Publisher to retrieve those keys as needed.

Users are given a username and password that they need to provide to Publisher to run it. This username and password is encrypted into the update itself to provide author tracking. All logins to the NOC are logged. Password strength rules are enforced.

6.10 Publisher

Publisher is provided with the netUpdate Agent as a tool to generate update description files, encrypt modules, and create an update "package" for a specified set of update components. Using Publisher you specify the input components (partitions, files, etc), and the OUI/Model Group of the target device. Using this information and the operator's login credentials Publisher contacts the NOC over an SSL connection to acquire the encrypt/decrypt keys., As part of the encryption process, Publisher pre-pends a RSA2048 encrypted key header with decryption instructions for every module being encrypted. The key header that Publisher builds includes the symmetric key for module decryption and a SHA256 decrypted module payload hash. Publisher appends a trailing signature to the end of the module payload data prior to encryption to verify the accuracy of the decrypted results. Large components are encrypted into smaller 2MB modules for transmission over the internet or broadcast networks.



6.11USB File Format

When updates are distributed by USB all of the modules that are normally transmitted one at a time on the internet or broadcast network are assembled into a single file with a pre-pended component directory that serves to glue them all together. The component directory is encrypted in exactly the same way as any other update payload is encrypted using a RSA2048 encrypted key header followed by a AES128 or HW-specific encryption of the main body of the component directory. This results in a tamper-proof encrypted update format that can be distributed publicly, placed on the internet, etc.

7 The Extdev and Production NOCs

UpdateLogic operates two NOCs to support netProvision. The extdev NOC is for development. The provisioned objects it contains are encrypted and protected using the protocols described in this document, but the devices they are being delivered to are assumed to be "open". Those devices may have serial input, telnet and other development and diagnostic communication channels enabled and may not employ secure boot. The provisioned objects imported to the extdev NOC are therefore considered to be for development only because they are at risk. The extdev NOC also allows universal ULPKs with an index of zero to be used to authenticate devices requesting provisioning. This is done to simplify development logistics. The extdev NOC supports all functionality that the production NOC does. It also supports a number of special development mode features that are not supported by the production NOC like limiting provisioned objects to a particular serial number range, etc.

The production NOC is used to deliver production provisioned objects and updates. It does not accept the universal ULPK. It is only for devices that are completely locked down. Because the provisioned objects served to these devices are production-ready the device must employ secure boot and not have serial input, telnet, or other diagnostic communication channels enabled.