|  |  |
| --- | --- |
| Input Contribution | |
| Meeting ID\* | SEC 31 |
| Title:\* | Use Case 5 - IoT application Certificate – Trusted Root |
| Source:\* | Ian Deakin, iconectiv, ideakin@iconectiv.com |
| Date:\* | 2017-09-21 |
| Input related to\* | WI-0073 - App-ID Registry Function |
| Intended purpose of  document:\* | Decision  Discussion  Information  Other <specify> |
| Impacted other TS/TR(s) | NA |
| Decision requested or recommendation:\* | Approve this section to the TR |
| Template Version: January 2017 (Do not modify) | |

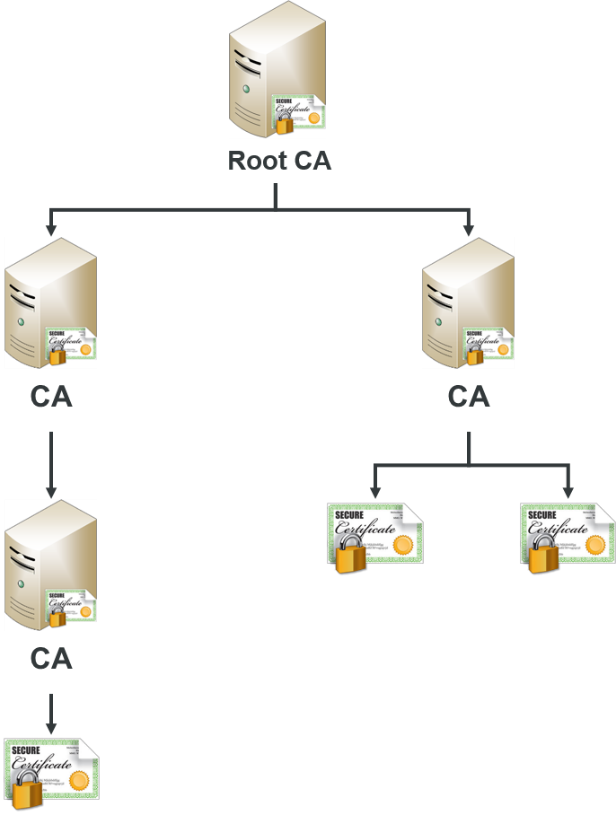
**oneM2M Notice**

The document to which this cover statement is attached is submitted to oneM2M. Participation in, or attendance at, any activity of oneM2M, constitutes acceptance of and agreement to be bound by terms of the Working Procedures and the Partnership Agreement, including the Intellectual Property Rights (IPR) Principles Governing oneM2M Work found in Annex 1 of the Partnership Agreement.

# 5.x Use Case : IoT application Certificate – Trusted Root

### 5.x.1 Description

A Certification Authority (CA) may either issue CA certificates or end entity certificates. A Root CA has a self-signed CA certificate and issues CA certificates to subordinate CAs. Trust in an end entity certificate requires trusting the Root CA.



**Figure 5.x.1-1: Certificate Hiarachy**

Digital certificates are stored in a certificate store. Major operating system (OS) vendors, Microsoft and Apple, include Root CA certificates in their Trusted Root Certification Authority certificate store that have met their security requirements. Some applications use the OS certificate stores, while others (e.g. Firefox and Oracle) have separate certificate stores.

Although OS and application vendors have a list of Trust Root CA certificates, additional Root CA certificates may be added to the Trusted Root Certification Authority certificate store by Administrators and end users with required security permissions. If a Root CA is compromised or no longer trusted, security patches must be applied that removes them from the Trusted Root Certification Authority certificate store or they must be removed manually.

**Real world Examples for the dangers of accepting a root certificates**

A recent Google warning over fake SSL certificates demonstrates, how one ‘rogue’ CA issuing unreliable certificates can cause havoc. Unfortunately, Certification Authorities can (and have been known to) issue fake certificates.

Trusting a malicious root is one of those nuclear-level “game over” scenarios. In fact, Chromium (the open-source project Google’s Chrome is based on) acknowledges that if an attacker can install a Root CA certificate onto your device, there is nothing the browser can do to protect you.

Current versions of Windows and OSX browsers provide notifications when a Root CA is not trusted. Trusting the Root CA requires manual installation of the Root CA certificate. For Windows, this takes quite a few steps, including viewing the Root CA certificate in the certificate chain and using the certificate import wizard.

On iOS, it is a different story. In Safari, just clicking a button on a webpage can prompt a system dialog to install a custom “profile” which can include Root CA certificates. An example of a profile that contains Root CA certificates is Comcast’s Xfinity Wifi. While iOS 10.3 and later does not automatically trust the certificate for SSL, earlier versions do trust the certificate.

User added Root CA certificates could be used maliciously. The new certificates could be used in a man-in-the-middle attack. This has always been a known vulnerability, but has not been of major concern because while feasible, it’s impractical in most attack scenarios.

**Managing trust and root certificates.**

Because of the growing variety of certificates in use today and the growing number of certificate issues, some organizations may want to manage certificate trust and prevent users in the domain from configuring their own set of trusted root certificates. In addition, some organizations may want to identify and distribute specific trusted root CA certificates to enable business scenarios where additional trust relationships are needed.

Adding a Root CA certificate to the Trusted Root Certification Authority certificate store can be quite labor intensive. The scale for IoT applications that could connect from a variety of issuing CA’s could be overwhelming for an organization. As indicated previously with the possibility of many IoT application developers using certificates from the same vendors, one root CA certificate for all could be dangerous, where as managing intermediate certificates for individual application vendors would be overwhelming.

**Using App-ID Registry Function as a trusted root for the Service provider infrastructure**

When an App-ID is registered, the metadata registered with the App-ID can contain information regarding the PKI structure, including:

* Issuing CA
* Intermediate CA certificate, if used
* Root certificate and its validity
* Location of the CRL ( certificate revocation list )

The SP infrastructure can query a connecting App-ID /AE-ID identity from the App-ID register function to collect the PKI metadata, to verify the authentication procedures of oneM2M ( MAF) .

* The SP infrastructure can verify that the Root CA certificate is valid and can be trusted (to what level) .
* That the issuing CA is valid and is still supported by the root CA.
* If an intermediate certificate is required to authenticate the IoT application certificate.
* Verify with the CRL that the certificate is not revoked.

The App-ID Registry Function can act as the root of trust for the oneM2M infrastryucture, enabling it to authenticate any registered IoT application through an automated process.