Mt Wilson

OIDs

# Background

Mt Wilson uses X.509 with custom attributes or standard data structures that have not been registered for OIDs (that we can find). New OIDs had to be assigned, and this document describes the OIDs in use by Mt Wilson.

# Tag Attributes

Mt Wilson uses two attribute structures for tag certificates.

## UTF8NameValueMicroformat

OID: 2.5.4.789.1

joint-iso-itu-t(2) ds(5) attributeType(4) mtwilson(789) name-value-microformat(1)

This OID is not registered.

The structure is a DER UTF8String whose value is of the form “name=value”.

Definition:

UTF8NameValueMicroformat ::= UTF8String

The format of the UTF8String is:

UTF8NameValueMicroformat ::= name “=” value

name ::= <any character except equal-sign(“=”)>

value ::= <any character>

## UTF8NameValueSequence

OID: 2.5.4.789.2

joint-iso-itu-t(2) ds(5) attributeType(4) mtwilson(789) name-value-sequence(2)

This OID is not registered.

The structure is a name and sequence of values.

UTF8NameValueMicroformat ::= SEQUENCE {

name UTF8String

values SEQUENCE OF UTF8STring

}

# TPM Certified Key Attributes

OID: 2.5.4.133.3.2.41

joint-iso-itu-t(2) ds(5) attributeType(4) tcg(133) tpm12(3) structures(2) certify-info(41)

This OID is not registered.

The OID 2.5.4 is parent to many X.509 attribute types.

# TPM Certified Key Signature

OID: 2.5.4.133.3.2.41.1

joint-iso-itu-t(2) ds(5) attributeType(4) tcg(133) tpm12(3) structures(2) certify-info(41) signature(1)

This OID is not registered.

The structure is the sequence of bytes generated by TPM\_CertifyKey function (ordinal 0x00000032) that represents the signature.

The OID 2.5.4 is parent to many X.509 attribute types.

# Roadmap

It might be necessary in the future to consolidate these under a single “Mt Wilson” OID, which would need to be registered, probably under Intel’s OID 1.3.6.1.4.1.343. From there we could request a new arc trusted-computing(8) tcg(33) tpm12(3) structures(2). Under that we could define each structure from the specification, for example tpm-sign-info(5) and tpm-certify-info2(41).

Alternatively, we can request registration of an OID for TPM structures under 2.5.4 which is what is used for many X.509 attributes. So from there we could request a new arc tcg(133) tpm12(3) structures(2). Under 2.5.4.133.3.2 we could define each structure from the specification, for example tpm-sign-info(5) and tpm-certify-info2(41). At the same time, tpm10(1) and tpm11(2) could be defined under tcg(133) as backward placeholders, while tpm20(4) could also be defined as a placeholder for those structures.

TCG actually has an OID 2.23.133 under which they have some certificate extensions but unfortunately they did not define all the data structures there so a third alternative is to request registration of the rest of the data structures under that arc.

The tcg(133) is in reference to TCG OID 2.23.133.

This registration was initially delayed because the authority listed in the OID registry (Adam Kaminsky) is no longer at Intel and because the current Intel representative was not immediately found, the matter was set aside.

The transition can be done in two phases: first, modify all our X.509 parsing/verifying code to accept the new OIDs, preferring new OIDs and falling back to the old OIDs, and inform any partners or customers with tight integration of the change; second, modify our X.509 generating code to use the new OIDs instead of the old ones.