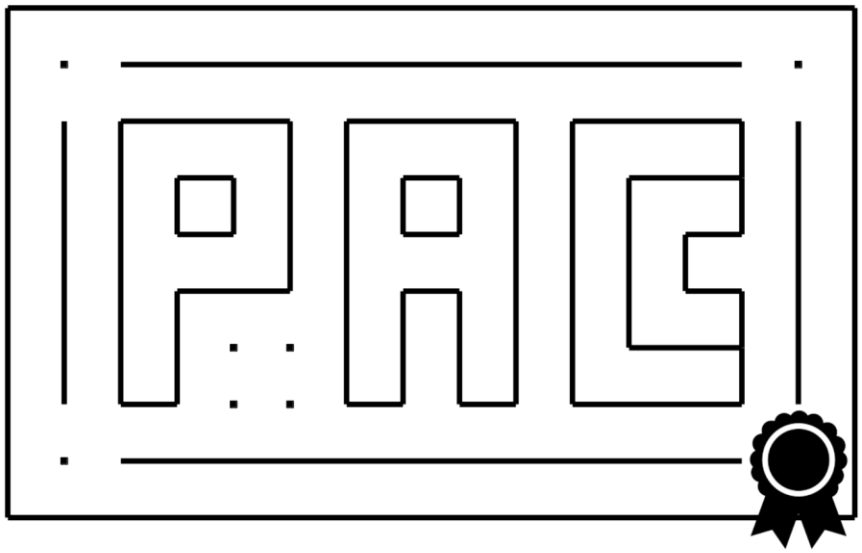

Platform Attribute Certificate Creator

Contents

Enable a trusted supply chain for computing devices

Abstract

PACCOR (1) is an open source tool for creating and testing Platform Certificates. A platform certificate encapsulates details about components on a host and the security standards met by the platform manufacturer. The certificate provides information to the customer and enables the mitigation of Supply Chain Risk Management concerns. It provides evidence of how the platform was built. Software can perform verification of that evidence against the platform to ensure no hardware components have changed anywhere along the supply chain from the factory to the customer's desk. The platform certificate is a 2KB file, and enables a supply chain validation capability that covers 100% of the devices on a network. Devices can be tested at any time to ensure they are still in the same state. The standards are open, and the proof of concept software is free and open-source. Read more for discussion on the impact of the platform certificate on customers as well as manufacturers, how the platform certificate is created, and why everyone benefits from platform certificate validation.

The Platform Certificate

Background

The platform certificate, or platform attribute certificate, is a X.509 attribute certificate. It is defined in a specification by the Trusted Computing Group, or TCG. "Platform Certificates contain assertions about trust made by a platform manufacturer." (2) The certificate contains a signature by the manufacturer to provide accountability to those assertions. The combination of those assertions and the signature makes the certificate an artifact of evidence which can be used to validate the platform. Verification of the assertions against the platform is meant to establish trust that the platform is in the same state as when the manufacturer signed the certificate. This means that the value of the certificate is directly proportional to how early in the production process the certificate is created. If the manufacturer creates the certificate at the factory, it can be used to validate the device at any point afterward. One goal of PACCOR is to ease the burden on manufacturers creating these artifacts, and especially on manufacturers to create these artifacts at the factory.

Related Specifications

The TCG is also responsible for two other specifications which are relevant to the platform certificate, and help explain its use. The first relevant specification is for the Trusted Platform Module, or TPM. The TPM is a cryptoprocessor that can perform cryptographic functions, including hashing, encryption, decryption, signing, verification, and key generation. (3) It can store data, most applicably to the platform certificate are keys and certificates. The second relevant specification is for the Endorsement Key (EK) Credential. "The EK is an asymmetric key pair…The public part of the EK can be read from the TPM while the private part must never be exposed. The public key of the EK in included in the EK Certificate." (4) The EK offers a way to identify a TPM and verify communication with a specific TPM.

Certificate Structure

X.509 attribute certificates use a DER-encoded ASN.1 structure. The platform certificate largely follows the structure of an attribute certificate as defined in RFC 5755 (5). The TCG has defined new fields to meet the functionality goals of the platform certificate. Since it is an attribute certificate, the platform certificate does not represent a distinct key pair. Instead, the platform certificate identifies an EK

certificate in its holder attribute. This relationship provides a way to bind additional attributes to the EK, such that when the validity of the TPM is established, those additional attributes have a trusted reference point.

Assertions made by the Platform Certificate

The platform certificate can be thought of a hardware bill of materials. It encapsulates details about components on a host and the security standards met by the platform manufacturer. A complete certificate would enumerate every hardware component on the system to include manufacturers, model numbers, serial numbers, and additional details which can be used to identify those components. The platform certificate profile (2) discusses all of the data it can contain. To give a brief list, the certificate carries:

- Device component details
  - i.e. the serial number for the motherboard
- Platform compliance to standards
  - Such as FIPS-140 Level 2
- One or more Endorsement Key References
  - To bind a TPM to the Platform
- A signature from the platform manufacturer
- and more…

Establishment of trust

The TPM is the root of trust. Confidence in the integrity of the TPM is required before anything else can begin to be trusted. That confidence can be increased by purchasing TPMs from trustworthy TPM manufacturers and ensuring it meets security requirements. These are exactly the details encapsulated in the EK Certificate, along with the public EK itself. Remember the private portion of the key is stored securely inside the TPM and is never exposed.

“An external entity attests to a TPM in order to vouch that the TPM is genuine and complies with [the] TPM specification. This attestation takes the form of an asymmetric key embedded in a genuine TPM, plus a credential that vouches for the public key of that pair.” (6)

The platform certificate is signed by the platform manufacturer to provide accountability.

“An external entity attests to a platform in order to vouch that the platform contains a Root-of-Trust-for-Measurement, a genuine TPM, plus a trusted path between the RTM and the TPM. This attestation takes the form of a credential that vouches for information including the public key of the asymmetric key pair in the TPM.” (6)

Successful validation of the EK certificate enables confidence in the TPM. Validation of the platform certificate is needed to gain confidence in the platform hardware.

Modifications after the factory, including by the customer

Legitimate changes may be made to the platform after it has left the factory. The platform certificate can accommodate those changes through the notion of a delta platform certificate. The base platform certificate would be the original certificate created at the factory by the original platform manufacturer. The base platform certificate reflects the state of the device when the first manufacturer associates a

TPM with the platform through an EK. Any manufacturer, value added reseller, distributor along the route may add or remove components to the platform. Each of those entities which modifies the platform will create a delta platform certificate, noting those modifications, and sign it with their own certificate authority. Validation of the complete set of base and delta platform certificates would succeed as long as the customer trusts the entities which performed the modifications. The customer itself may want to make changes to the platform. In this case, the customer, or an office the customer trusts, would perform the modification, and create a new delta platform certificate to attest to those changes. Again, as long as the entity that performed the modifications to the platform is trusted, the validation will succeed.

PACCOR can help create both base and delta platform certificates.

Utilization

The platform certificate only has meaning after it has been signed. Therefore, the platform certificate should be created as early in the production process as possible. If the platform manufacturer creates the platform certificate on the production line at the factory, then the manufacturer, and any entity down the supply chain to the customer, may use that certificate to ensure the platform is still in the same state.

Validation of the platform certificate starts with verifying the current date is within the certificate's validity period, and verifying that the certificate was signed by a trusted certificate chain. Each of those certificates represents a key pair owned by different levels of one or more organizations. This means the each entity in the supply chain must provide the chain of certificates used to sign their platform certificate.

Next, the holder, referenced by a distinguished name and serial number, is validated. For a base platform certificate, the holder is the EK certificate stored in the TPM. For a delta platform certificate, the holder is either a base or delta platform certificate. This means that a chain of platform certificates reflect changes to the platform along the supply chain. Ultimately one or more EK certificates are referenced and validated to ensure the TPM and all of its EKs are worthy of trust.

Validation continues by verifying all of the attestations made within the platform certificate. All components of the platform at any stage of the supply chain should be enumerated in the platform certificate. Software can compare what is in the certificate with what is seen on the system. If all of the components and security assertions check out, then supply chain validation of the hardwares is successful.

Reasons why the platform manufacturer should care

Counterfeit and swapped components affect manufacturers as well. If a customer receives a product with a counterfeit component from a trusted manufacturer, and that component is not detected, performance or security could be compromised for the customer. Empowering the customer to detect that counterfeit component proves that the manufacturer cares about the customer and ensures the manufacturer is not distributing compromised components.

The byte size and effort to create the platform certificate is not unreasonable. I believe this capability can be designed such that creation of these certificates should not have any efficiency cost on the

production line. The security gained for supplying the certificate is great- to the point that it should be a no-brainer.

There are strong benefits to manufacturers even if they don't agree that this technology is worth pursuing. Platform Certificates are not the only technology which would benefit from teaching vendors how to integrate certificate and key management into their build process. Certs and keys enable extraordinary trust mechanisms for the customer. If the customer is purchasing a device with a TPM, they would benefit from having as much verifiable information as practical. Open standards, like the platform certificate, provide information in a uniform manner. As other standards are identified, the manufacturer would be ready to enhance or replace existing standards.

Reasons why the customer should care

Customer would be able to detect counterfeit, relabeled, missing, extra, and removed hardware components on their system as soon as they receive the box. The test occurs at runtime, and takes less than 1 minute to perform. The platform certificate empowers the customer to verify they received exactly what they purchased.

Customers should start requesting that all new computing devices they buy come with a TPM, an Endorsement Key, and a Platform Certificate. Those three elements provide a way to establish trust in the system hardware. As standards evolve, additional verification of firmware and software is enabled because the underlying hardware can be trusted.

Benefits and Drawbacks for implementing

First, the benefits:

- Enables detection of counterfeit, relabeled, missing, extra, and removed hardware components on the system at any time from factory to desk.
- The Standards are open.
- The capability requires 3 artifacts to be delivered with the system, 2 of which are generally available today- that is the TPM and the EK certificate are widely available today. The platform certificate is the next step to gain assurance about more than the TPM.
- Tools are available to support creation of the platform certificate.
- Each platform certificate is roughly 2KB in size.
- Effect on Production line efficiency should be minimal if any at all.

Below are multiple drawbacks for the platform certificate, and discussion about those drawbacks. Please go to the Feedback section to provide additional benefits or drawbacks as they are encountered. We want to be the biggest critic of this technology. If an unsurmountable vulnerability is found with the platform certificate, we want to know about it.

Dependencies for automated validation

Automated validation relies on the machine to report accurate data. There are no standards yet on what system tools will provide the best source for data to compare against the platform certificate. PACCOR provides examples on how to pull that data from the system. It demonstrates using SMBIOS and lshw for Linux, WMIC and CMIv2 for Windows. This means that those tools must be checked by the manufacturer to ensure they report accurate data that reflects what is encapsulated into the platform certificate. Starting with v1.1 of the platform certificate, component data can be labeled as coming from

a particular source. If those labels are used, particular focus must be kept on ensuring those tools are given proper data which will match the certificate.

Next, sometimes the data within these sources is mutable. This is where privileges on the device must be maintained properly by everyone involved in the supply chain. Part of having a manufacturer's certificate in a customer's trust chain is the fact that the customer trusts the manufacturer. Once that trust is lost, the customer may revoke that certificate, and all machines on the customer's network which has a component from that manufacturer will cease to validate. The signature on the platform certificate carries accountability for those assertions. There is a financial and reputational incentive for every actor in the supply chain to act in good faith to providing secure and complete products. The goal of the trusted supply chain is not to question that trust, but to empower everyone the opportunity to verify their device meets agreed upon requirements.

String base comparisons

Related to the dependencies drawback is the fact that the comparison method for component details and source data is by String. String comparisons have multiple hurdles to overcome, because text fields can contain any amount of additional characters, including spaces and commas. More than just ensuring the source data is complete is also verifying that data is formatted in a consistent manner. This will enable validation tools to perform these string comparisons correctly.

Acceptance with established tools

Attribute certificates are not widely adopted by available open source cryptography tools. OpenSSL, for instance, has problems processing attribute certificates. Attribute certificates are defined in RFC 5755. (5) They are included in the ITU.T X.509 framework. (7) The lack of existing tool support could be due to a lack of public interest. As the community sees value in attribute certificates, existing tools will meet that necessity. Similarly, as interest in TCG standards rise, those tools will see demand for implementing the new structures proposed in those standards. Until then, new tools, like PACCOR, can fill the role of simplifying the implementation of these new standards.

Virtualization

The TPM can be implemented as a hardware chip soldered onto the motherboard. There are emulators and efforts to virtualize the TPM. Discussion on which of these methods are preferred is not in the scope of this guide. The TPM is the root of trust of any capability presented in this guide.

Privacy Concerns

There are privacy concerns about EK usage and they are covered in the EK specification (4). The purpose of the TPM, the EK, and the platform certificate is to enable the system owner to build trust in their system. There are protections in place to prevent access to those artifacts if the system owner does not know about, or does not want to expose, those artifacts. This guide is meant to discuss the opportunities for greater platform security through the use of the platform certificate.

Components not visible by the system

Hardware components could be added to the system which are hidden or not registered by tools like SMBIOS. They are essentially invisible to the system. The manufacturer may not include them in the platform certificate. This can be the result of malice or absent-mindedness. The platform certificate cannot aid in detection of those components. Validation of the platform hardware could succeed despite that hidden component. The device may be trusted to be placed on the network.

Performing this trusted supply chain validation is still critical, because it will create records of all machines on the network. Those records will document manufacturers, model numbers, serial numbers of each system as well as each component. If any component is found to be compromised, those records will identify every machine on the network which possesses that model component. They will identify every manufacturer that had a part in creating the system. The records can be used during forensic analysis to determine where the problem originated and how much hardware must be removed from the system.

Trust in the TPM and EK

Trust in the platform certificate crumbles in the event that the TPM, or the Endorsement Key stored within it, are determined to be not trustworthy. All three components are open standards, which means their specifications are subject to peer review and discussion. Much of the software implementations of them are also open-sourced and subject to code review. This greatly reduces the chance that there are security vulnerabilities with any of them. It does not eliminate that chance. This is also true of all of the algorithms used within each specification. This is the nature of computer science. Peer review, code review, and good faith sharing of lessons are the best defenses for hidden vulnerabilities.

PACCOR

Goals

PACCOR is a tool for creating platform certificates. The three main goals of the project are:

1) To ease the burden on entities who wish to create platform attribute certificates according to the specification.
    a. This is accomplished by the Java programs
2) To demonstrate how details from a platform can be gathered in a quick, automated manner
    a. This is accomplished by the scripts

Quick Start

PACCOR is available on GitHub. (1)

PACCOR targets the TCG Platform Certificate Profile v1.1. (2)

The program consists of 3 tools and a set of platform-dependent scripts. The tools are written in Java and are intended to be platform agnostic. The tools should work the same on any platform. They require Java 1.8 or greater. Each tool has usage details available on the command line via -h. Gathering platform details is inherently platform-specific. That work is intended to be handled by platform-dependent scripts. PACCOR has scripts which support Linux and Windows. More details on those are available in the Scripts section.

A number of the command line options for PACCOR Java tools request data formatted in JSON. Any data gathered by the scripts are output into JSON for ingestion into the Java tools. Descriptions for the expected format of that data is available in the JSON Structures section.

Signer

Validator

Observer – intended to work as an agent so that details can be gathered from one platform and delivered to another machine which hosts the Attribute Authority creating the platform certificates.

PACCOR Releases are the easiest place to start. There is a script pc_certgen, which will generate a platform certificate for your system. You can then compare the output files against your system to see how it is reflected in the certificate.

Programs

Signer

This tool creates the X.509 v2 Attribute Certificate according to the Platform Attribute Certificate Profile. It will sign the certificate with the given private key. Data requested from the profile can be given to the tool via JSON. The JSON Structures section describes the format. Scripts are provided to reduce that work to only filling in a few variables. The signing algorithm is chosen by the signing public key certificate. See

Appendix III: PACCOR Supported Signing Algorithm OIDs.

Arg	Alternate Arg	Format
-c	--componentJsonFile	See Component JSON
	the path to the file with the Device Info JSON	Use allcomponents.sh
-p	--policyRefJsonFile	See Policy Reference JSON
	the path to the file with the Policy Reference JSON	Use referenceoptions.sh
-e	--holderCertFile	Formatted in DER or PEM
	for a base platform certificate, the X.509 v3 EK Certificate	Use get_ek.sh
	for a delta certificate, reference another platform certificate	
-N	--serialNumber	Integer
	serial number for the new certificate	
-b	--notBeforeDate	YYYYMMDD
	valid not before date	
-a	--notAfterDate	YYYYMMDD
	valid not after date	
-k	--privateKeyFile	Formatted in DER or PEM
	the private key used for signing the certificate	
-P	--publicCertificate	Formatted in DER or PEM
	the public key certificate corresponding to the private key	
-x	--extensionsJsonFile	See Extensions JSON
	the path to the file with the Extensions JSON	Use otherextensions.sh
-f	--out	Optional field
	the output file path. If not set, stdout will be used.	
	--pem	Optional field
	change output to PEM format. DER format is default	
-h	--help	
	prints the help message to stdout	

Validator

This tool will validate the signature of a X.509 v2 Attribute Certificate. In the future, this validator could be enhanced to verify other claims within the attribute certificate.

Arg	Alternate Arg	Format
-P	--publicCertificate	Formatted in DER or PEM
	the public key certificate of the signing key	
-X	--attributeCertificate	Formatted in DER or PEM
	the X.509 Attribute Certificate with the signature to validate	
-h	--help	
	prints the help message to stdout	

Observer

The intention of this device observer tool is to provide a simple way to gather certificate data locally from a device and transfer it to a central location where the signer tool can perform its job. The output from the observer tool can be given directly to the signer.

Arg	Alternate Arg	Format
-c	--componentJsonFile	See Component JSON
	the path to the file with the Component JSON	Use allcomponents.sh
-p	--policyRefJsonFile	See Policy Reference JSON
	the path to the file with the Policy Reference JSON	Use referenceoptions.sh
-e	--holderCertFile	Formatted in DER or PEM
	for a base platform certificate, the X.509 v3 EK Certificate	Use get_ek.sh
	for a delta certificate, reference another platform certificate	
-f	--out	Optional field
	the output file path. If not set, stdout will be used.	

TCG ASN.1 Library for Java Bouncy castle

Bouncycastle provides a great cryptographic library which supports attribute certificates. (8) PACCOR includes extensions for Bouncycastle to support TCG-defined ASN.1 structures.

Scripts

allcomponents

This script queries the operating system to find facts regarding hardware components on the device. It looks at SMBIOS, and also looks at OS-specific tools. The script should require no user interaction beyond running it.

- Platforms Supported
  - Linux: in particular CentOS 7 and Ubuntu 18.04
    - Requires superuser privileges due to some OS queries
      - Does not alter the system at all
  - Windows
    - Does not require superuser, Windows allows those OS queries by all users

referenceoptions

This script is intended to condense platform manufacturer-defined Platform Attribute Certificate data fields into a few variables. The user should open the script and edit the variables to specify what standards are met by the platform model. Each variable below will reference the profile (2) for additional details.

Note: SHA-256 was chosen to be acceptable for each of the hashAlg choices for URI references. That can be changed by advanced users by editing this script.

- Platforms Supported
  - Linux
    - Does not require special user privileges
  - Windows
    - Does not require special user privileges

Variable	Value	Reference
tcgPlatformSpecificationMajorVersion	Number	TCG Platform Specification Attribute
tcgPlatformSpecificationMinorVersion	Number	
tcgPlatformSpecificationRevision	Number	
platformClass (9)	Number	
tcgCredentialSpecificationMajorVersion	Number	TCG Credential Specification Attribute
tcgCredentialSpecificationMinorVersion	Number	
tcgCredentialSpecificationRevision	Number	
platformConfigUri	URI	Platform Configuration Uri Attribute
platformConfigLocalCopyForHashing	Path	URI References require a hash of the file
tbbSecurityAssertionVersion	Number	Default: 1; Platform Assertions Attribute
commonCriteriaMeasuresVersion	String	see reference publications at https://CommonCriteriaPortal.org/cc
assuranceLevel	Number	valid options are 1 thru 7
evaluationStatus	Enum	valid options: designedToMeet, evaluationInProgress, evaluationCompleted
ccPlus	Enum	default false; valid options: true, false

strengthOfFunction	Enum	valid options: basic, medium, high
profileOid	ObjectId	OID of the protection profile
profileUri	URI	Platform Assertions Attribute
profileLocalCopyForHashing	Path	URI References require a hash of the file
targetOid	ObjectId	OID of the protection target
targetUri	URI	Platform Assertions
targetLocalCopyForHashing	Path	URI References require a hash of the file
fipsVersion	String	see reference publications at https://csrc.nist.gov/Projects/Cryptographic-Module-Validation-Program/Standards
fipsLevel	Number	
fipsPlus	Enum	default false; valid options: true, false
measurementRootType	Enum	valid options: static, dynamic, nonHost, hybrid, physical, virtual
iso9000Certified	Enum	default false, valid options: true, false
iso9000Uri	URI	Simply a String in v1 of the spec.

otherextensions

This script provides access to variables which can likely be set once by an attribute authority. In particular, the Certificate Policies, the CRL distribution list, and authority key access extensions can be defined using these variables. The user should open the script and edit the variables for their purpose. Each variable below will reference the certificate profile for additional details.

Note: SHA-256 was chosen to be acceptable for each of the hashAlg choices for URI references. That can be changed by advanced users by editing this script.

- Platforms Supported
  - Linux
    - Does not require special user privileges
  - Windows
    - Does not require special user privileges

Variable	**Value**	**Reference**
certPolicyOid1	ObjectId	Certificate Policies
certPolicyQualifierCPS1	String	Certificate Policies
certPolicyQualifierUserNotice1	String	Specific value defined in the profile
authorityInfoAccessMethod1	Enum	valid options are OCSP or CAISSUERS
authorityInfoAccessLocation1	DN	Authority Info Access
crlType	Number	valid options are 0 or 1
crlName	DN	CRL Distribution
crlReasonFlags	Number	valid options are integers 0 thru 16
crlIssuer	DN	CRL Distribution

get_ek

This script will attempt to gather the EK certificate from a TPM.

- Platforms Supported
    - Linux, especially CentOS 7 and Ubuntu 18.04
        - Requires superuser privileges
        - Should work for TPM v1.2 or v2.0
    - Windows 10
        - Requires administrator privileges
        - Should work for TPM v1.2 or v2.0

pc_certgen

This script automates the entire process to use PACCOR with the goal of demonstrating each step of the process to create a platform certificate. It produces a platform certificate signed by a newly generated key pair. All settings can be customized. The generated key pair can be replaced with the user's specific key pair.

- Platforms Supported
    - Linux, especially CentOS 7 and Ubuntu 18.04
        - Requires superuser privileges due to calls to allcomponents and get_ek
        - Subsequent calls do not require superuser privileges
    - Windows 10
        - Requires administrator privileges due to call to get_ek
        - If EK certificate is provided out of band, no other calls require admin

JSON Structures

Component JSON

```
{
    "PLATFORM": {
        "PLATFORMMANUFACTURERSTR": "",
        "PLATFORMMANUFACTURERID": "",               // optional
        "PLATFORMMODEL": "",
        "PLATFORMVERSION": "",
        "PLATFORMSERIAL": ""                        // optional
    },
    "COMPONENTS": [
        {                                           // zero or more component objects
            "COMPONENTCLASS": {
                "COMPONENTCLASSREGISTRY": "",
                "COMPONENTCLASSVALUE": ""
            }
            "MANUFACTURER": "",
            "MANUFACTURERID": "",                   // optional
            "MODEL": "",
            "FIELDREPLACEABLE": "",                 // optional
            "SERIAL": "",                           // optional
            "REVISION": "",                         // optional
            "ADDRESSES": [                          // optional, zero or more of these 3 address types
                {
                    "ETHERNETMAC": ""
                },
                {
                    "WLANMAC": ""
                },
                {
                    "BLUETOOTHMAC": ""
                }
            ],
            "PLATFORMCERT": {
                "ATTRIBUTECERTIDENTIFIER": {        // optional
                    "UNIFORMRESOURCEIDENTIFIER": "",// required if ATTRIBUTECERTIDENTIFIER is included
                    "HASHALGORITHM": "",            // optional
                    "HASHVALUE": ""                 // optional, base64 encode the value
                },
                "GENERICCERTIDENTIFIER": {          // optional
                    "ISSUER": "",                   // DN
                    "SERIAL": ""
                }
            },
```

Policy Reference JSON

```
{
    "TCGPLATFORMSPECIFICATION": {
        "VERSION": {
            "MAJORVERSION": "",
            "MINORVERSION": "",
            "REVISION": "",
        },
        "PLATFORMCLASS": ""                     // hex value, base64 encode the value
    },
    "TCGCREDENTIALSPECIFICATION": {
        "MAJORVERSION": "",
        "MINORVERSION": "",
        "REVISION": ""
    },
    "TBBSECURITYASSERTIONS": {
        "VERSION": "",
        "ISO9000CERTIFIED": "",
        "CCINFO": {                             // optional
            "VERSION": "",
            "ASSURANCELEVEL": "",               // optional
            "EVALUATIONSTATUS": "",             // optional
            "PLUS": "",
            "STRENGTHOFFUNCTION": "",           // optional
            "PROFILEOID": "",                   // optional
            "PROFILEURI": {                     // optional
                "UNIFORMRESOURCEIDENTIFIER": "",// required if PROFILEURI included
                "HASHALGORITHM": "",            // optional
                "HASHVALUE": ""                 // optional, base64 encode value
            },
            "TARGETOID": "",                    // optional
            "TARGETURI": {                      // optional
                "UNIFORMRESOURCEIDENTIFIER": "",// required if TARGETURI included
                "HASHALGORITHM": "",            // optional
                "HASHVALUE": ""                 // optional, base64 encode value
            }
        },
        "FIPSLEVEL": {                          // optional
            "VERSION": "",
            "LEVEL": "",
            "PLUS": ""
        },
        "RTMTYPE": "",                          // optional
```

Extensions JSON

```
{
    "CERTIFICATEPOLICIES": [
        {
            "POLICYIDENTIFIER": "",
            "POLICYQUALIFIERS": [
                {
                    "POLICYQUALIFIERID": "",
                    "QUALIFIER": ""
                },{
                    "POLICYQUALIFIERID": "USERNOTICE",
                    "QUALIFIER": "TCG Trusted Platform Endorsement"     // fixed value
                }
            ]
        }
    ],
    "AUTHORITYINFOACCESS": [
        {
            "ACCESSMETHOD": "",
            "ACCESSLOCATION": ""
        }
    ],
    "CRLDISTRIBUTION": {
        "DISTRIBUTIONNAME": {
            "TYPE": "",
            "NAME": ""
        },
        "REASON": "",
        "ISSUER": ""
    },
    "TARGETINGINFORMATION": [                                          // zero or more target EKs
        {
            "FILE": ""
        }
    ]
}
```

What's next

I will update this document with experiences as I learn of manufacturer and customer lessons in implementing platform certificates.

Implementation

Ensure dependencies are properly informed

Ensure SMBIOS reports consistent, complete, trustworthy component details. The platform manufacturer should confirm that every component they document into a platform certificate can be verified against data reported by the platform. The customer should perform an acceptance test on any device they purchase to ensure that the platform certificate validates their new device.

Create your AA, or CA

Each organization that builds or makes alterations to a platform must document those changes in a platform certificate. The authority used to sign the platform certificate is called a certificate authority (CA), or more accurately called an attribute authority (AA) for an attribute certificate. The private key used for each layer of the CA must be protected as it is used to provide accountability and confidence in the assertions of the platform certificate. The platform manufacturer would need their own CA, as would any value-added-reseller. Even the enterprise will need their own CA to issue platform certificates, because system owners may want to make hardware configuration changes.

Produce Platform Certificates at the factory

PACCOR can help. As components are being added to the system, there might be a way to start building the component list that goes into the platform certificate in real time. At the end of the line, the policy and CA-level details are added to the certificate as it is signed. Finally a check of the system verifies for the platform manufacturer that the platform certificate can be used to validate the hardware configuration of the system. I believe any efficiency impact on the production line should be minimal. If I can help work through problems with implementing this at the factory, let me know via Feedback. The TCG is another place that problems can be worked through. Their contact information is on their website.

Methods for delivery of Platform Certificates

The TPM has NVRAM space for the platform certificate. For TPM v2.0, there is no specific NVRAM index recommended, rather a range of indexes for Platform objects. They can be placed in the range of 0x01C08000 to 0x01C0FFFF. (10) Delivery on the TPM is preferable due to always having access to that base certificate. The certificate could also be provided out of band to the customer, perhaps via the web or on a disk that accompanies the product.

Methods for delivery of Trust Chains

Certificate chains should be made available via the internet. There should be no concern about providing intermediate to root public key certificates for this context. Multiple TPM manufacturers make their EK certificate chains available on their websites. Platform manufacturers could distribute their certificates in a similar way.

Use cases for intermediary creation of delta platform certificates

During the supply chain process multiple vendors may look at or modify a platform. Value added resellers may add additional memory or remove a hard drive. Those changes must be documented in

delta platform certificates so that the customer can verify the state the platform was in at the factory and at every stage of delivery to their desk.

Validate early, re-validate often

Establish trust in every device bought as early as possible, preferably before connecting it to central networks. 100% of devices can be tested at any time. The ACA can help (11).

Feedback

Many implementation details are untested because this is a newer capability. We have focused on understanding the underlying technology and making software tools available. We would be happy to have an opportunity to hear about experiences or offer assistance with implementing trusted supply chain. This feedback would be valuable with perspectives from any role: from the OEM, to system integrators, to the customer. HIRS@tycho.nsa.gov

References

1. **NSACyber.** Platform Attribute Certificate Creator. *GitHub.* [Online] [Cited: April 17, 2019.] https://www.github.com/nsacyber/paccor.

2. **Trusted Computing Group.** Platform Certificate Profile v1.1r15 Public Review. *Public Review Specifications.* [Online] February 13, 2019. https://trustedcomputinggroup.org/wp-content/uploads/IWG_Platform_Certificate_Profile_v1p1_r15_pubrev.pdf.

3. —. TPM 2.0 Library Specification. [Online] September 29, 2016. [Cited: April 15, 2019.] https://trustedcomputinggroup.org/resource/tpm-library-specification/.

4. —. TCG EK Credential Profile v2.1r13. *Specification.* [Online] December 10, 2018. https://trustedcomputinggroup.org/wp-content/uploads/TCG_IWG_Credential_Profile_EK_V2.1_R13.pdf.

5. **Farrell, S., et al.** RFC 5755 - An Internet Attribute Certificate Profile for Authorization. *Internet Engineering Task Force (IETF).* [Online] January 2010. [Cited: April 15, 2019.] https://tools.ietf.org/html/rfc5755.

6. **Trusted Computing Group.** Trusted Platform Module Library Part 1: Architecture v1.38. [Online] September 29, 2016. [Cited: April 15, 2019.] Section 9.5.3.1. https://trustedcomputinggroup.org/wp-content/uploads/TPM-Rev-2.0-Part-1-Architecture-01.38.pdf.

7. **International Telecommunication Union.** ITU-T X.509. *ITU-T Recommendation Database.* [Online] [Cited: April 15, 2019.] https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=X.509.

8. **Bouncy Castle.** Bouncy Castle Java Cryptography APIs. [Online] [Cited: April 20, 2019.] https://www.bouncycastle.org/java.html.

9. **Trusted Computing Group.** Registry of Reserved TPM 2.0 Handles and Localities v1.1r1. *Reference.* [Online] February 6, 2019. Table 13. https://trustedcomputinggroup.org/wp-content/uploads/RegistryOfReservedTPM2HandlesAndLocalities_v1p1_pub.pdf.

10. —. Registry of Reserved TPM 2.0 Handles and Localities v1.1r1. *Reference.* [Online] February 6, 2019. Table 2. https://trustedcomputinggroup.org/wp-content/uploads/RegistryOfReservedTPM2HandlesAndLocalities_v1p1_pub.pdf.

11. **NSACyber.** Host Integrity at Runtime and Startup Attestation Certificate Authority. *GitHub.* [Online] [Cited: April 17, 2019.] https://www.github.com/nsacyber/hirs.

Appendix I: Assumptions

Not Specified

Currently, SMBIOS is sometimes not well informed about the details of all of the components on the system. Sometimes a platform serial number might be empty. In the platform certificate, if a component field is required, and the OS responds with an empty string, the allcomponents script will insert the String “Not Specified”. This is because the field is required, and having an empty string for that field is ambiguous. This is a demonstration choice and is not necessarily the same choice that would be made by all vendors. Over time, SMBIOS and all dependencies of the platform certificate will have a complete set of data.

Appendix II: Table of component list sources

TO BE FILLED IN

Appendix III: PACCOR Supported Signing Algorithm OIDs

		OID
RSA	SHA1	1.2.840.113549.1.1.5
	SHA256	1.2.840.113549.1.1.11
	SHA384	1.2.840.113549.1.1.12
	SHA512	1.2.840.113549.1.1.13
DSA	SHA1	1.2.840.10040.4.3
	SHA256	2.16.840.1.101.3.4.3.2
	SHA384	2.16.840.1.101.3.4.3.3
	SHA512	2.16.840.1.101.3.4.3.4
ECDSA	SHA1	1.2.840.10045.4.1

Appendix IV: Glossary

ASN.1	Abstract Syntax Notation One
DER	Binary format of ASN.1 structures
EC	X509v3 Certificate representing the EK
EK	Endorsement Key
IWG	Infrastructure Working Group
JSON	JavaScript Object Notation
PACCOR	Platform Attribute Certificate Creator
PC	Platform Certificate
PEM	Base64 Encoded DER Certificate
TCG	Trusted Computing Group
TPM	Trusted Platform Module