psacertified™

Entity Attestation Token White Paper

How Service Providers can Trust the Internet of Things

Version: 1

psacertified™

Author: Arm

Date of Issue: 2020

Non-Confidential Proprietary Notice

This document is protected by copyright and other related rights and the practice or implementation of the information contained in this document may be protected by one or more patents or pending patent applications. No part of this document may be reproduced in any form by any means without the express prior written permission of Arm. No license, express or implied, by estoppel or otherwise to any intellectual property rights is granted by this document unless specifically stated.

Your access to the information in this document is conditional upon your acceptance that you will not use or permit others to use the information for the purposes of determining whether implementations infringe any third-party patents.

THIS DOCUMENT IS PROVIDED "AS IS". ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, Arm makes no representation with respect to, and has undertaken no analysis to identify or understand the scope and content of, patents, copyrights, trade secrets, or other rights. This document may include technical inaccuracies or typographical errors. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ARM BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

This document consists solely of commercial items. You shall be responsible for ensuring that any use, duplication or disclosure of this document complies fully with any relevant export laws and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word "partner" in reference to Arm's customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

If any of the provisions contained in these terms conflict with any of the provisions of any click through or signed written agreement covering this document with Arm, then the click through or signed written agreement prevails over and supersedes the conflicting provisions of these terms. This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of the Agreement shall prevail.

The Arm corporate logo and words marked with ® or ™ are registered trademarks or trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All rights reserved. Other brands and names mentioned in this document may be the trademarks of their respective owners. Please follow Arm's trademark usage guidelines at http://www.arm.com/company/policies/trademarks.

Arm Limited. Company 02557590 registered in England.
110 Fulbourn Road, Cambridge, England CB1 9NJ.

Entity Attestation Token White Paper

How Service Providers can Trust the Internet of Things

Abstract

As the digital world has changed and adapted, we have found ourselves with many connected devices that are quite different to the typical laptop or phone that we are used to. To make use of these new devices, Cloud Service Providers (CSPs) need to integrate them into their cloud platforms and require a way to identify, characterize and authenticate them. The usual Internet protocols and services for security and authentication are not device-oriented and do not suit this purpose. This white paper introduces the idea of device attestation, the upcoming Entity Attestation Token (EAT) standard and how the PSA Certified ecosystem is planning to support it.

Table of Contents

Terminology

EAT	Entity Attestation Token, a proposed Internet standard for attestation
CBOR	Compact Binary Object Representation. See http://cbor.io and https://www.rfc-editor.org/info/rfc7049.
ECDSA	Elliptic Curve Digital Signature Algorithm
COSE	CBOR Object Signing and Encryption [COSE.RFC]
JOSE	Javascript Object Signing and Encryption [JOSE.RFC]
JSON	Javascript Object Notation. See http://json.org
Relying Party	The server-side application that receives an EAT token and takes action based on its contents.
TF-M	Trusted Firmware-M, an open source software project for Arm Cortex-M processors.
IETF	Internet Engineering Task Force, publisher of RFC series of Internet standards
RFC	Request For Comment, a series of widely used Internet standards published by the IETF.
PSA	Platform Security Architecture, a security architecture for IoT devices.
Claim	A name-value item of data in an EAT, CWT or JWT that claims something about the device such as its ID, manufacturer or status.

Introduction

Historically, the Internet threat model focused on attackers along the communication path and solutions focused on offering communication security to mitigate these threats. Devices were assumed to be trustworthy and not compromised. As hackers got more sophisticated and the software ecosystem got more complex, threats evolved against the devices themselves. In addition to the introduction of anti-virus and malware detection software, sophisticated software isolation technologies, like Trusted Execution Environments (TEEs), were introduced. Internet of Things (IoT) devices introduce additional challenges because they typically need to be remotely managed via an IoT platform provider either because they do not have a user interface or because they are installed in inaccessible locations. Offering information about the device in form of attestation information allows these IoT devices to securely introduce themselves to networks and to IoT platforms.

The goal of device or attestation is therefore to provide information about the device to other parties using a very simple, cryptographically secured token. Each information item in this token in known as a 'claim' with some common examples given in Table 1.

What is an Entity Attestation Token (EAT)?

An Entity Attestation Token (EAT) is a small blob of data that contains claims and is cryptographically signed. The cryptographic signing secures the token so that the means to convey the token does not have to provide any security. For example, it can be transmitted by HyperText Transfer Protocol (HTTP), Constrained Application Protocol (COaP) or added into the extension fields of any protocol that supports extensions. EATs will often be conveyed in systems that use TLS for their own security needs, but TLS is not necessary for EAT. Typically, the Elliptic Curve Digital Signature Algorithm (ECDSA) is used for signing and the token size is a few hundred bytes for the claims and signature.

The claims that are contained in an EAT help the cloud-based service assess and decide whether to trust the device and its data. This "trust signal" could be used for example to decide whether to carry out a transaction the device is requesting.

Claim Name	Claim Description
Unique identifier	Similar to a serial number. Universally and globally identifies each individual device.
Manufacturer and model	Identifies the manufacturer of the chip and/or the finished device.
Installed software	Lists the software present on the device including versions.
Device boot and debug state	Indicates if the device booted securely, whether debug mode is enabled, and debug ports disabled.
Geographic position location	For example, based on GPS, WiFi, cell tower or some combination. Only available if the device has location features.
Versions, measurements and/or integrity checks of running software	Measurements of running software, usually hashes of the code, are provided for comparison against known-good-value to help detect tampering.
Nonce	Cryptographic quality random number generated, sent by the server and returned as a claim to prevent replay and reuse.

Table 1: Some common, useful claims used by Entity Attestation Tokens (EAT).

How Does EAT Relate to the Platform Security Architecture?

The Platform Security Architecture (PSA) is a framework that improves IoT security by applying a four-step approach to IoT: security analysis, architecting, implementation and certification. PSA defines an on-chip security component that provides trusted functions to the device and acts as a trust anchor for services known as the PSA Root of Trust (PSA-RoT). An open source software project known as Trusted Firmware-M (TF-M) provides a reference implementation of the following trusted functions: trusted boot, secure storage, cryptography libraries and attestation. EAT is seen as an important security asset going forward, it has been selected as the reference attestation method of PSA and TF-M, effectively becoming a default security service of the Root of Trust (RoT). The open source software for EAT can be found at www.trustedfirmware.org. IoT chips that support EAT can be found on the PSA Certified website displaying the PSA Functional API Certified logo (shown in Fig. 1). EAT is in the process of becoming an industry standard through work in the IETF, FIDO Alliance and GlobalPlatform.

Fig. 1: IoT products that support EAT can be identified by the PSA Functional API Certified logo.

Why Do We Do Device Attestation?

In a typical scenario, the device, which is often an IoT device (for example a sensor or an appliance) or a mainstream device (like a phone or even a laptop), provides an attestation to a server or service so that it can securely know characteristics of the device. The server or service is known as the relying party.

Device attestation can give a Cloud Service Provider (CSP) critical information it needs to decide whether to onboard an IoT device, particularly those that have no user account associated with them. Critical information includes:

- Enabling the relying party to know the type and manufacturer of the device hardware, including the main chip in the device. Attestation provides this information with cryptographic proof that is resistant to forgery. Devices attempting to forge this identification will not be able to do so because they do not have the necessary key material.
- The relying party can know the security configuration of the chip, including if the device booted securely and if debug ports are disabled.
- The relying party can know the specific individual instance of the device from the UEID, which is similar to a serial number. In many implementations the UEID will be set in hardware, it will not be possible for the end user to change it.
- The relying party can know what software is running on the device and that it has not been tampered with.

Use cases for device attestation include:

Manufacturing

A branded business may not manufacture goods themselves but instead use a contract manufacturer who will directly ship devices to the retail outlet. Attestation allows the devices of several such businesses to be strongly distinguished.

Network requirements

An enterprise may wish to limit devices it allows on its network, or limit access to higher-value data, by verifying that the devices requesting access are known to have specific levels of security.

Utility protection

When data provided by a device is of high criticality, for example from autonomous driving, medical devices or the power grid, attestation provides assurance that it is coming from an authentic device and is not forged.

Financial trading

When processing financial transactions, the claims in the attestation may be used as high-quality inputs to a risk engine to enable better risk management.

It is important to note that not every implementation will provide all of these benefits. Early implementations will likely provide fewer. Some will not be provided because privacy regulations disallow them.

Anatomy of an Entity Attestation Token

The core of an EAT is the set of claims, the core is then surrounded by the structure to sign and secure the claims.

Tokens can use either CBOR or JSON format to express the claims (but not a mixture in one token). JSON is used widely in web services. CBOR is a binary size-efficient format.

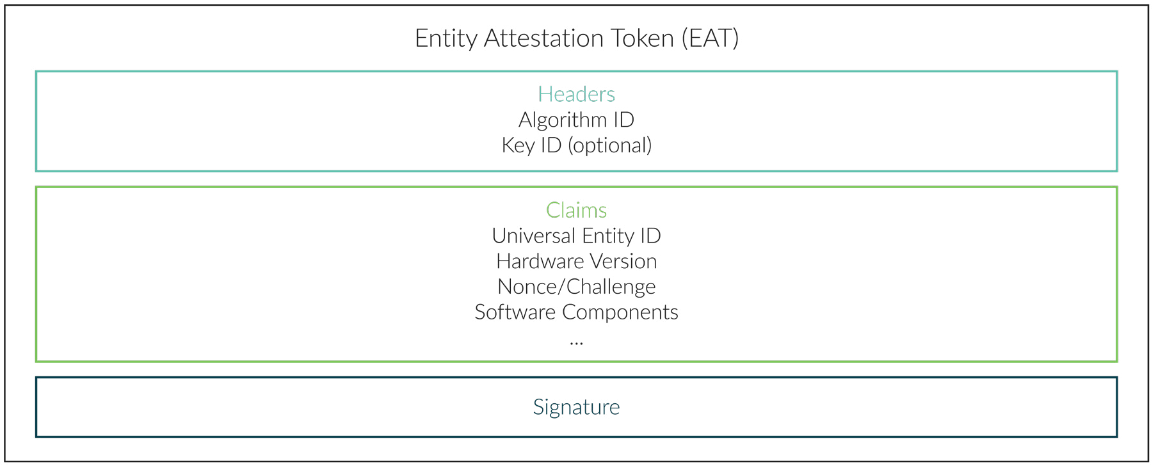

Fig 2. Pictorially shows an example of an EAT, depicting both CBOR and JSON formatted tokens.

Claims

The claims section is a set of label-value pairs, whereby each claim has a label and a value. In CBOR the labels are typically simple small integers and in JSON they are strings. The claim values can be as simple as an integer or as complex as nested arrays with members of many types.

The EAT standard defines a set of specific claims that are expected to be common to many use cases. In the EAT standard, all claims are optional, but profiles of the standard will make some claims mandatory. Bespoke claims are allowed.

Internet Assigned Numbers Authority (IANA) will operate a worldwide registry of defined claims. A use case that needs a particular type of claim can check there to see if one has been previously defined and reuse that claim, rather than inventing their own. If there is no such claim, they may invent their own – upon inventing their own claims, they can decide whether the claim is private, should be publicly documented and registered, or whether to go through the process to standardize it.

Signing Structure

The modern COSE and JOSE standard signing formats are used. They can work with many different signing algorithms and key types. The headers identify the signing algorithm and optionally the signing key. This provides futureproofing as signing algorithms and key lengths change over time. The signature is at the end of the token and gives integrity protection to the claims.

COSE is based on CBOR and is particularly size-efficient. For a 256-bit ECDSA signature the size of the headers and the signature is less than 90 bytes, most of which is the actual ECDSA signature itself.

Relation to CBOR Web Token (CWT) and JSON Web Token (JWT)

EAT is considered to be a variant of CWT or JWT. All the requirements and characteristics of CWT and JWT apply to EAT. All the claims defined for these can be used as claims in EAT. The IANA registry is to be shared across these token types.

Software libraries that implement CWT, JWT, COSE and JOSE can be reused to implement EAT, with little modification in some scenarios.

How Does Device Attestation Work?

The Attestation Service could be operated by the chip/device manufacturer, a third-party Attestation Service Provider, or a CSP.

The Relying Party is the end consumer of the attestation result. It makes the decision to trust the device or not for whatever purpose or transaction is in progress.

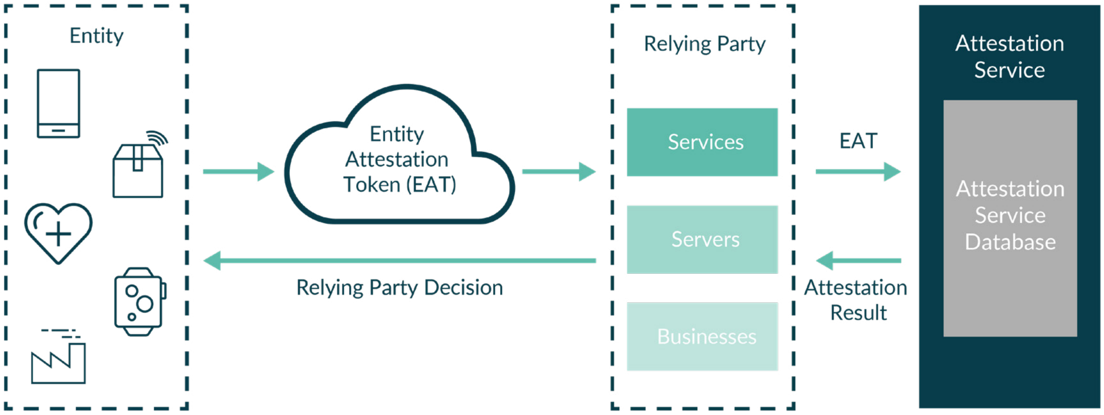

Fig.3: A high-level view of how device attestation works.

Step 1 – Key Set Up

Attestation typically uses a signing algorithm like ECDSA. In this step, the private part of the ECDSA key is put into a secure location in the device or chip, usually the RoT. This key will be used to sign the attestation token and the private part is kept secret so that the relying party knows the signature could only have come from an authentic device. This is the fundamental basis for attestation security.

Correspondingly, the public part of the ECDSA key goes into a database housed by the attestation service. This will be used to verify the signature on the attestation token.

The public and private keys are linked by a key ID. They key ID is put into the device alongside the private key. It is also alongside the public key in the attestation service database.

The provisioning of the private key is usually done in a secured step in the chip/device manufacturer's factory. It is done only once in the life of the device. Usually, a unique key is put into each device.

Using a PSA-RoT, the key is hardware isolated and has a level of protection from software or hardware attack. This security is critical in making the system more secure as it relies on the attacker not getting this private key as it would allow them to forge tokens from inauthentic devices.

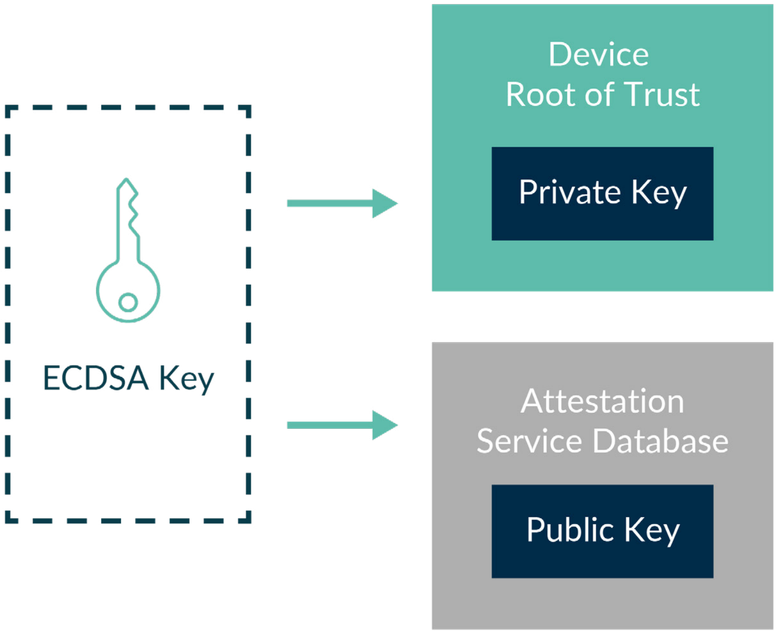

Fig 4. During key set up, the private part of the key is housed in the device RoT and the public part of the key is housed in the attestation service database. They are linked by a key ID.

Step 2 – Generating an EAT Token

In this step, an EAT token is generated and the claims are gathered. The ECDSA private key is used to sign the claims and it is put together in the correct format for an EAT token.

One of the claims included is a nonce or challenge to prevent replay or re-use of the token. This is a small sequence of random bytes that is sent from the attestation service to the device.

The TF-M Initial Attestation implementation is trusted firmware that can be protected by isolation technologies. PSA supports high level, easy-to-use APIs to the Root of Trust functions making it easy for developers to generate a token.

EAT token generation is performed many times, perhaps for each transaction the chip/device is involved with. It is relatively inexpensive because the claims data is small and ECDSA signing is relatively fast.

Once formatted the token is sent off the device. Typically, it will go to the relying party who then relays it to the attestation service for verification. As it passes through, the relying party will not examine or modify it: they merely relay it.

The tokens are small, 300-600 bytes in the TF-M implementation, and self-secured by signing. They can be put into an HTTP header or into the extension of some other protocol already in use for conveyance from the device to the relying party. HTTPS or TLS is not necessary.

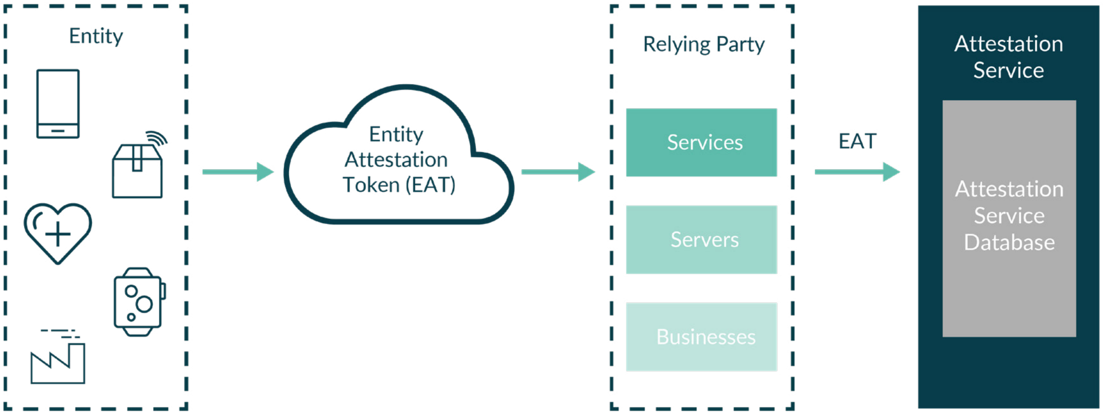

Fig 5. The EAT token is generated and sent to the relying party who then relays it to the attestation service for verification.

Step 3 – Verification by the Attestation Service

One likely design of the attestation service is that of a web API that takes in a token and outputs an attestation result. The API is secured by HTTPS so the relying party knows the token is being verified by a trustworthy verifier and so the results are not tampered with.

When the verifier receives the token, it uses the key ID to look up the public key. Then it verifies the signature to establish that the claims were not tampered with in transit and that they came from a trusted device.

At this point some verifiers will just return the verified claims to the relying party. Other verifiers may validate some of the claims. For example, if the claims include version and measurement information about the software on the device, they will check that against known-good-values.

The data returned by the attestation service to the relying party is called the Attestation Result.

This document does not assume any particular ecosystem for the attestation service. It is presented in a generic way. In practice, it might be operated by the chip or device vendor themselves. The service could also be operated by a CSP, an interested third party, an operating system vendor or an industry organization. It is even possible that the relying party does the verification internally without the use of an attestation service, though in this case they would have to obtain the public keys from the device manufacturer.

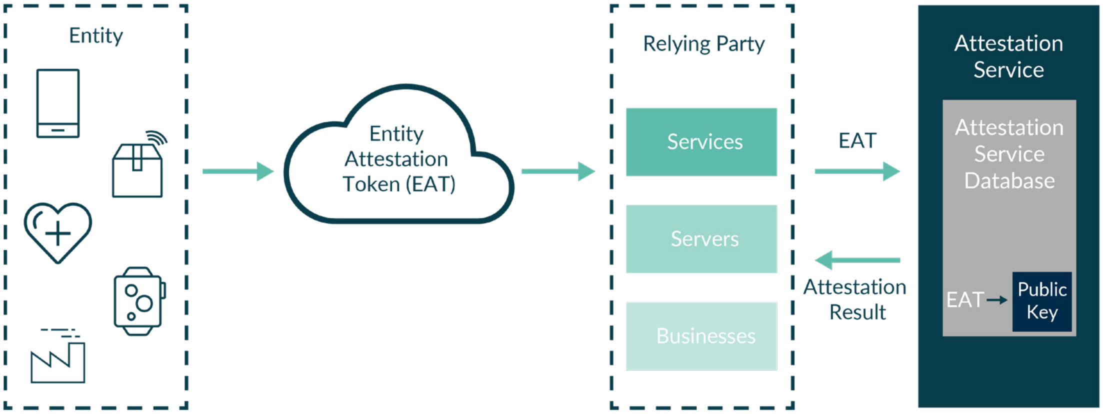

Fig 6. The EAT token is generated and sent to the relying party who then relays it to the attestation service for verification.

Step 4 – Relying Party Decision

The verifier is typically generic for many different types of device and use cases; it does not know what service is being requested by the device. However, the relying party does because it is responsible for carrying out the service, for example, it is responsible for carrying out a payment transaction.

The attestation result received from the verifier can be very simple (for example: approved/not approved or yes/no), or it could be more complex including all the claims from the original token. Furthermore, it could include information about the device that the attestation service can infer by knowing the type and manufacturer of the device.

The relying party will use the attestation result and perhaps some local context or transaction-specific information to decide if it will continue. For example, it may or may not decide to onboard the device depending on the manufacturer of the device. For another example, it may decide to allow a financial transaction if the amount is modest and the device is considered modestly trustworthy and the transaction is not out of the ordinary for the user in question.

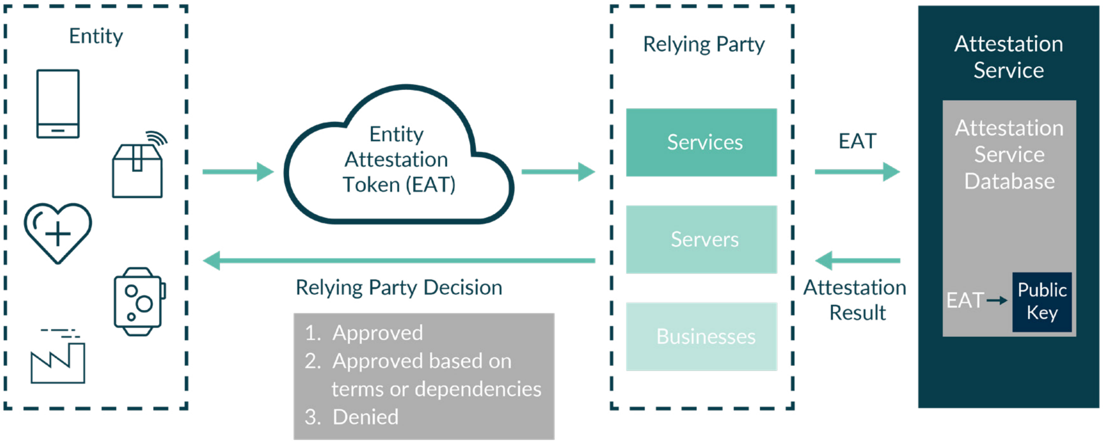

Fig 7. The relying party will use the attestation result to make transaction-specific decisions.

Use Case Descriptions

Trust Signal for Cloud Services

A cloud service may wish to know a set of claims about the device to make risk judgements about it – is it one of my managed group and does it need an update? Has its security been independently evaluated and if so at what level? Which version of the software is running? Did it boot securely?

IoT Onboarding

An IoT device may be made in a contract manufacturer's factory and drop-shipped into the retail supply chain such that the official manufacturer of the device, the one offering some back-end services for it, never sees the device. The contract manufacturer is given ECDSA keys to put in the device by the manufacturer. When the device pops up on the network to register with the manufacturer's service, the service knows it is a legitimate device and not one from a competitor.

Payment Transactions

A service clearing payment transaction may want to know that the payment transaction originated on a secured device as part of its risk management strategy. By examining an attestation token from the device that comes along with the transaction request, the payment service can know the manufacturer of the device and whether it is configured correctly.

Enterprise Access

An enterprise or government may wish to restrict which types of devices are allowed on a private network or have access to servers with high-value data. Similar to the payment example, the enterprise can examine the token and know the manufacturer and state of the device.

Remote Monitoring

A very secure electric metering implementation could be based substantially on an attestation token. The token could include the following data items:

a. Device serial number
b. GPS location
c. Electricity used
d. Tampering indicators

Since they are all signed by the ECDSA key they are all bound together so the power company can know the device is in the right geographic place and has not been tampered with.

IoT Device Authentication

A maker of an IoT device may offer a web service associated with the device that the consumers who have purchased the device can use. For example, a toaster vendor may have a website to which the consumer can upload images to be burned into the toast that it makes. Attestation can be used to prevent competing toaster vendors from using their service by checking the signed and secured device vendor identification in the EAT.

Standardization

In spring 2019, the IETF Remote Attestation Procedures (RATS) working group was formed to create an Internet standard for attestation. EAT was adopted for standardization by this working group and it has been progressing.

EAT is based on CBOR Web Token (CWT) or JASON Web Token (JWT). JWT is widely deployed with RESTful APIs on the web and with smartphone apps. JWT is also the preferred encoding of tokens with OAuth. CWT is a more recent specification, but widespread deployment is also anticipated. The core means for structuring, serializing and signing claims are largely settled. See pre-standard versions of EAT here.

A major focus of the RATS work is to define new claims for device attestation. These fit in easily next to the already-defined authentication-oriented CWT and JWT claims.

CWT and JWT use IANA registries to track the definitions of different claims which can be found respectively here and here. EAT will do the same. The process allows for claims that are fully standardized by an organization, claims that are not standardized but have public specifications and private and vendor-specific claims.

EAT, CWT and JWT use the COSE (CBOR Object Signing and Encryption) and JOSE (JSON Object Signing and Encryption) proposed standards to sign and secure the claims. These are modern formats for data signing and encryption. They provide long-term cryptographic agility as new algorithms replace old ones over periods of decades. COSE is a good fit for IoT because it is compact and efficient.

The EAT standard is intended to be very general and span a very large set of use cases. Standardized profiles for more specific classes of use cases are expected to be developed. A standardized profile may specify which claims are allowed and which are required. They are expected to standardize on specific signing algorithms and key sizes. For example, there may be one profile for attestation in Trusted Execution Environments (TEE) in mobile phones, another for common household appliances and another for high-security grid infrastructure.

There are also attestation standards work in the FIDO Alliance and GlobalPlatform, both generally in harmony with the IETF work. These organizations are likely to create some standard EAT profiles.

EAT is supported as part of Trusted Firmware-M. The PSA Certified program has an API compliance program called PSA Functional API Certified that requires and covers EAT implementations. OEMs and service providers can find chips and RTOS that support EAT on the PSA Certified website.

Summary

Building a trustworthy IoT is one of the biggest issues facing the technology industry today. EAT can help solve this problem by providing attested claims, anchored in the Root of Trust (RoT), that builds trust between the device and cloud services.

There is a big advantage in converging on one attestation format as it will allow service providers to process attestation tokens from multiple vendors in a uniform way, rather than having to implement a different format (or have no attestation at all) for each device manufacturer.

If you are interested in using EAT in your next IoT project you can find chips and RTOS that support EAT and the PSA Functional APIs on the PSA Certified website. If you want to integrate EAT in your next chip or RTOS, you can find an open source implementation as part of the Trusted Firmware–M (TF-M) project. Cloud service providers or operators interested in doing a Proof of Concept are welcome to contact the PSA Certified team, using PSAcertified@arm.com.

Find out More

PSA Certified is a comprehensive assurance scheme for securing digital transformation across industries. The PSA Certified founders and key industry support work on the scheme to ensure it adds value and aligns the market requirements. Offering peace of mind, trust and a lower cost of ownership. Putting security at the heart of your product.

Find out more at psacertified.org