WHITE PAPER

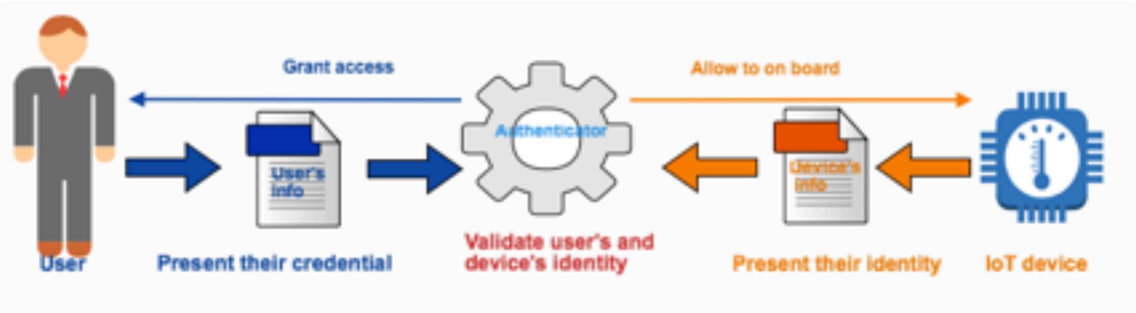
AUTHENTICATION AND AUTHORISATION FOR IOT APPLICATION

Having a secure platform and environment for IoT devices to operate is important, in order to ensure safety of users and prevent unwanted consequences that might occur as a result of a security loophole. In this sense, organisations delivering and using IoT systems must be diligent in their defence of device data and security aspects must be considered and implemented in their early design phase, not as an afterthought.
Sirris has built up extensive experience and knowhow in IoT security and will publish a series of white papers in which all the issues will be discussed in detail.

Our second white paper focuses on authentication and authorisation for IoT application. In this paper, we look at different authentication and authorisation methods, the existing standard authentication and authorisation protocols and the challenges.

Authentication and authorisation, processes of identifying an individual and giving individual access to system objects based on their identity, play an important role in providing protection to IoT system objects against attacks. These processes ensure that devices connected to systems, as well as the associated application users, are genuine and prevent malicious devices and users from tampering with the system. Given their strategic importance, it is crucial to properly select the right methods suitable to the needs, use case and context of the application.

sirris
driving industry by technology



1. DEVICE & USER AUTHENTICATION

Deploying a robust authentication system is the first step of providing security and considered as the backbone for any security strategy. It will ensure a much safer browsing experience and online data exchange. In the context of IoT system, two main authentication strategies need to be considered when designing IoT-based applications:

a) Device authentication is a process of identifying an individual device wanting to onboard the system or join the network. This process aims at ensuring that only a genuine device is allowed to be in.

b) While device authentication ensures only a genuine device is allowed to join the network, user authentication makes sure that only a genuine user is able to access the system, devices and data.

Authentication in IoT is not as simple as in a resource-abundance system. This is because IoT deals mostly with constrained devices, which have less computing power and memory to handle complex authentication processes designed for less constrained devices. The authentication methods designed for non-IoT application are completely unsuitable for an IoT environment or they need to be extended in order to be useful for IoT. Below are some authentication methods that can be used in IoT, but with different levels of security.

1. **Identify-based**. Authentication is based on the unique identity of a user or device, for instance, a user's biometric. Biometric authentication methods include retina, iris, fingerprint and finger vein scans, facial and voice recognition.

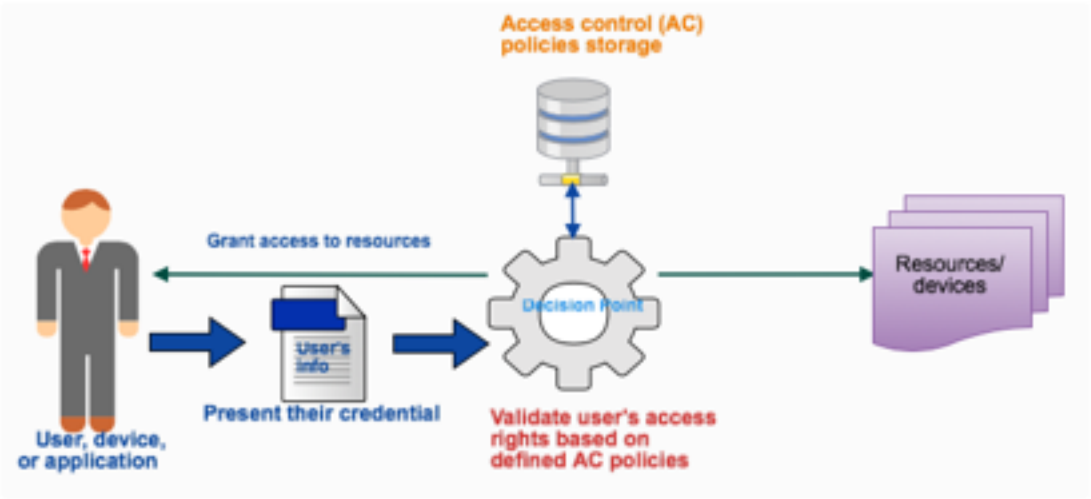
2. **Username & Password** is a traditional authentication method that has been used since the early history of computing. However, the days of one-step authentication with a username and password are over. A more secure method such as MFA is required in order to move organizations away from a high-risk password-based security model.
3. **Multi-factors authentication (MFA)** is an authentication method in which a device or application user is granted access only after successfully presenting two or more pieces of evidence to an authenticator. For example, Two-factors authentication where a user needs to provide (1) something they know (e.g. username and password) and (2) something they have (e.g. random generated number).
4. **Token-based authentication** is a security technique that authenticates the users who attempt to log in to a server, a network, or some other secure system, using a security token provided by the server. The token acts like an electronic key to access something. Some tokens may store cryptographic keys, such as a digital signature, or biometric data. The token can be used in addition to or in place of a password.
5. **Context-aware authentication** is a complex and advanced authentication method. It combines the existing authentication methods in conjunction with the contextual information (e.g. location, time, ...) to enforce the authentication process.
6. **Cryptographic authentication** is a method allowing a user or device to identify themselves based on the cryptographic keys. Either symmetric key or public key authentication.
7. **Mutual authentication**, also known as two-way authentication, is a process in which both entities in a communication link authenticate each other. In IoT context, the user of the application and the IoT device can use mutual authentication to prove the identity of each other in the network, before allowing the user accessing the device.

Existing standard authentication protocol. There are some well-known industrial standard authentication protocols (see table below) that can be used in IoT context. Many more are also available in the research literatures [1]. However, precaution should be taken seriously if non-standard protocol is adopted in the development of IoT application.

Name	Type	Description
OAuth	Token-based	**OAuth** is a token-based authentication and authorisation open standard for internet communications. OAuth is not designed specifically for securing IoT, which has a few unique caveats that need to be considered, such as battery powered, connectivity and limit computing power. However, OAuth can be extended to support authentication in IoT. For example, a proof-of-possession method, where OAuth token is simply used to prove device or user identity "we are who we say we are".
MQTT	Username and passwords	MQTT [9] is a lightweight protocol often used for devices to communicate with other systems. It is designed for the publish/subscribe messaging pattern. The MQTT protocol supports a basic authentication mechanism based on usernames & passwords.
HTTP Basic	Username and passwords	In this approach, an HTTP user simply provides a **username** and **password** in HTTP header to prove their authentication. The issue is that, unless the process is strictly enforced throughout the entire data cycle to SSL for security, the authentication is transmitted in open on insecure lines. This lends to man-in-the-middle attacks, where a user can simply capture the login data and authenticate via a copy-cat HTTP header attached to a malicious packet.
CoAP	Token-based	CoAP [8] is an Internet Application Protocol designed for constrained devices. CoAP is a one-to-one protocol for transferring state information between client and server. MQTT and CoAP are both useful as IoT protocols but have fundamental differences. MQTT is a many-to-many communication protocol for passing messages between multiple clients through a central broker. It decouples producer and consumer by letting clients publish and having the broker decide where to route and copy messages.
OpenID connect	Token-based	OpenID Connect is a simple identity layer on top of the OAuth 2.0 protocol. It allows clients to verify the identity of the end-user/device based on the authentication performed by an authorisation server, as well as to obtain profile information about the end-user.
SASL	Depend on authentication mechanism	Simple Authentication and Security Layer (SASL) is a framework for authentication and data security in Internet protocols. It separates authentication mechanisms from application protocols, in theory allowing any authentication mechanism supported by SASL to be used in any application protocol that uses SASL.

Selection of the authentication protocol. There are a number of authentication methods proposed in literature [1][2] as well as the mature industrial standard ones [3]. Many can be used or extended to be useful for an IoT environment. When it comes to the selection of an authentication method for a particular IoT application, it all depends on the actual requirements of the application use case and an in-depth study of requirements of the application is required. However, preliminary selection can be performed by looking at the following parameters (see table below).

Best practice: select the right authentication method	
Constrained vs less constrained devices	It is important to know the capacity (memory and computing power) of the envisioned devices (sensors/actuators and the client's devices) used in the system, since the capacity of the device can affect the selection of the authentication method. For example, with a less constrained device, OAuth can be used while with a constrained one, MQTT or CoAP may be more suitable.
Security: critical vs normal system	The selection of the authentication method depends also on the criticality of the envisioned system or application itself. The IoT system for mobile healthcare is more critical than air quality control for a smart home. The balance between security and usability should also be considered.
Communication technology and protocol	Some authentication methods are designed to function with a particular communication protocol. For example, OAuth works specifically under HTTP. Thus, the selection of the authentication method depends also on the use of a communication protocol. Some communication protocols support limit data exchange capability that does not support a complex authorisation method with heavy processes.


Access control (AC) policies storage
Grant access to resources
User's info
Decision Point
Resources/ devices
Present their credential
Validate user's access rights based on defined AC policies
User, device, or application

2. AUTHORISATION FOR IOT APPLICATION

Authorisation is the process of granting permissions on specific actions to given entities - specifically to users, devices, or applications. Who can access and use which object in which circumstances? Generally, authorisation is about controlling the access rights of an individual to system resources (e.g. devices and data shared in the system). There are different authorisation models ranging from a simple identity-based to a more fine-grained authorisation model such as RBAC, ABAC or UCON [5]. A fine-grained model generally requires complex policy validation processes, hence, more computing power and memory. This model tends to be less suitable for constrained IoT devices.

There are many access control models [4][5]. However, we discuss here only some well-known models that can be used in the context of IoT. These models have been extensively studied, developed and some incorporated into industrial standard applications.

1. Discretionary access control (DAC) is a type of access control that restricts access to objects (or resources) based on the identity of subjects and/or groups that they belong to. The controls are discretionary in the sense that a subject with a certain access permission is capable of passing that permission directly or indirectly on to any other subject.
2. Attribute-based access control (ABAC). In ABAC, access rights are granted to users through the use of policies which combine attributes. The policies can use any type of attributes, such as subject attributes, object attributes or environment attributes. This model supports Boolean logic and allows policies to express a complex Boolean rule set that can evaluate many different attributes.
3. Role-based access control (RBAC) is a method of restricting access of users to objects based on the roles of individu-

al users within an organization. RBAC is a policy neutral access control mechanism defined around roles and privileges. RBAC lets users have access rights only to the information assigned to the role that belongs to them. RBAC can be used to facilitate large scale administration of security in large organizations with hundreds of users and thousands of permissions. RBAC has be studied extensively and as a result RBAC has many extensions [6].

4. Context-aware access control is a method of restricting access users to objects based on the contextual information. This model provides tight, just-in-time permissions so that authorized users get access to specific objects according to the current context. These permissions are subject to continuous adjustments triggered by the changing context, for instance, a temporal context.

Existing standard authorisation protocol. There are many authorisation protocols proposed in research literatures. However, a standard authorisation protocol developed specifically for an IoT application environment is not known to exist. Although there is no IoT-specific authorisation protocol, some existing protocols, such as OAuth and CoAP can support authorisation in some IoT contexts. Below is a list of authorisation protocols that can be used for IoT devices as well as application user authorisation.

Name	Description
OAuth 2.0	OAuth 2.0 supports not only authentication, but also authorisation. OAuth 2.0 is the industry-standard protocol for authorisation. It allows a user to grant limited access to their resources without having to expose their credentials.
SAML	Security Assertion Markup Language (SAML) is an open standard for exchanging authentication and authorisation data between parties, in particular, between an identity provider and a service provider. SAML is an XML-based markup language, similar to XACML, for expressing access-control decisions.
Delegated CoAP Authorisation Function (D-CoAP)	D-CoAP provides user authentication and authorisation in a constrained environment for establishing a Datagram Transport Layer Security (DTLS) channel between resource-constrained nodes. The protocol relies on DTLS to transfer authorisation information and shared secrets for symmetric cryptography between entities in a constrained network.
WS-Federation (WS-authorisation)	The WS-Authorisation is used to describe how access policies for a Web service are specified and eventually managed. The goal is to describe how claims can be specified within security tokens and how these claims will be interpreted at the endpoint.
XACML-based authorisation	XACML [7] stands for "eXtensible Access Control Markup Language". The standard defines a declarative fine-grained, attribute-based access control policy language, an architecture, and a processing model describing how to evaluate access requests according to the rules defined in policies.

Selection of the authorisation protocol. Some parameters need to be considered when selecting authorisation method. They relate to security of device, security of user, as well as, data circulating in the network. Which authorisation method should we choose? It all depends on the following factors.

Best practice: select the right authorisation method	
Required level of granularity of authorisation policies	The choice of the authorisation method is based on the level of control required. The fine-grain authorisation policies require a complex verification and validation processes, which tend to be less suitable for the constrained environment.
Organization structure	A complex configuration for user and resource may also affect the selection of authorisation model. For example, in large scale system with complex organization structure (e.g. devices or users), RBAC may be a good choice.
Device capacity	Some authorisation methods, for instance OAuth works specifically under HTTP. Constrained device may not have sufficient memory and computing power to support HTTP processes. Thus, the capacity of the device is also an important factor when it comes to the selection of an authorisation method.
Authorisation system: centralized vs distributed architecture	A distributed architecture, where the authorisation process is done at the device, can avoid a point of failure that generally happens in a centralized architecture. However, it also has a drawback compared to a centralized architecture. In a distributed architecture, it may not be possible to use a fined-grain authorisation method given that IoT devices are generally resources-constrained, hence, they do not support heavy verification and validation processes.

3. AUTHENTICATION AND AUTHORISATION IN IOT - CHALLENGES

The main challenge has to do with authenticating constrained devices. How to authenticate constrained devices with highly secure authentication mechanism, which usually involves complex process and high computing power? IoT devices, with limited computing power and memory, are not designed to handle heavy computing tasks. This means that standard authentication protocols, that involve heavy computing, used for less constrained devices do not work. Lightweight authentication is a great choice when it comes to authentication within IoT, due to the above-mentioned constraints. However, lightweight protocol may not be able to provide the same security level as the normal standard authentication protocol.

In general, to be able to select the right authentication and authorisation methods for a given application, it is important to conduct an in-depth study of the actual needs of the application. There is not one size fit for all. Many factors, as presented above, can affect the selection of authentication and authorisation methods. However, the rule of thumb for providing secure authentication and authorisation is as follows:

- Do not communicate authentication/authorisation information/data in an IoT environment in plain text without protection. A simple cryptographic algorithm at the least should be used to ensure basic security.
- Always authenticate application user and devices with secure authentication methods. In the situation where only authentication with password can be used, a strong password policy should be adopted. Determining the identity of device and user is also a part of the authentication process. Thus, a strong identity protection mechanism should be used, especially for devices (e.g. sensor or actuator).

- Since one-step username and password authentication is no longer safe, use MFA where possible, especially when it comes to authenticating the application user. This is because, the application user tends to use a less constrained device to connect to the system.
- For authorisation, fine-grained access control policy should be used where possible. Fine-grained policy provides a different level of control to data and devices and protect the identity, as well as the credentials of the user or device from exposition to the outside-world .

SUMMARY

The purpose of this paper is to provide an overview of what we can do concerning authentication and authorisation in an IoT environment. In the next instalment in this series, we will discuss in detail the communication security. The focuses will be on different communication protocols and technologies used in IoT environment.

REFERENCES

[1]. Michal Trnka , Tomas Cerny , and Nathaniel Stickney. Survey of Authentication and Authorisation for the Internet of Things. Security and Communication Networks Volume 2018, Hindawi, 17 pages.

[2]. Shancang Li. IoT Node Authentication. Securing the Internet of Things. Elsevier, 2017.

[3]. OAuth 2.0 & OpenID. https://oauth.net/2/ & https://openid.net/connect/

[4]. Hokeun Kim and Edward A. Lee. Authentication and Authorisation for the Internet of Things. IEEE digital library, IT Professional. September-October, 2017, pp. 27-33, vol. 19.

[5]. RBAC, DAC, ABAC. USA National Institute of Standard and Technology. https://csrc.nist.gov/publications/detail/nistir/7316/final.

[6]. Ni, Qun and Bertino, Elisa and Lobo, Jorge and Brodie, Carolyn and Karat, Clare-Marie and Karat, John and Trombeta, Alberto. Privacy-aware Role-based Access Control. ACM Trans. Inf. Syst. Secur. July 2010.

[7]. XACML. https://www.oasisopen.org/committees/tc_home.php?wg_abbrev=xacml

[8]. CoAP. https://coap.technology

[9]. MQTT. http://mqtt.org

AUTHOR

Annanda Rath

Senior Expert Software Engineering & Security

CONTACT : +32 493 31 06 45 - annanda.rath@sirris.be

Over the last 8 years, Annanda has been active in software and data security (in particular the access and usage control). At Sirris, he is a senior software security expert and actively working on different aspects of software security, from data to system security and privacy. He is particularly interested in data integrity and confidentiality in distributed environment: distributed ledger technology (e.g. blockchain and directed acyclic graph). Cloud, IoT and smart mobility are also his area of interest.

Annanda has a Phd in Computer Science from the University of Namur, a Master's degree in Computer Science from the Indian Institute of Technology, Bombay, India (IITB) and an Engineering degree in Computer Science from Cambodia Institute of Technology (ITC), Cambodia.