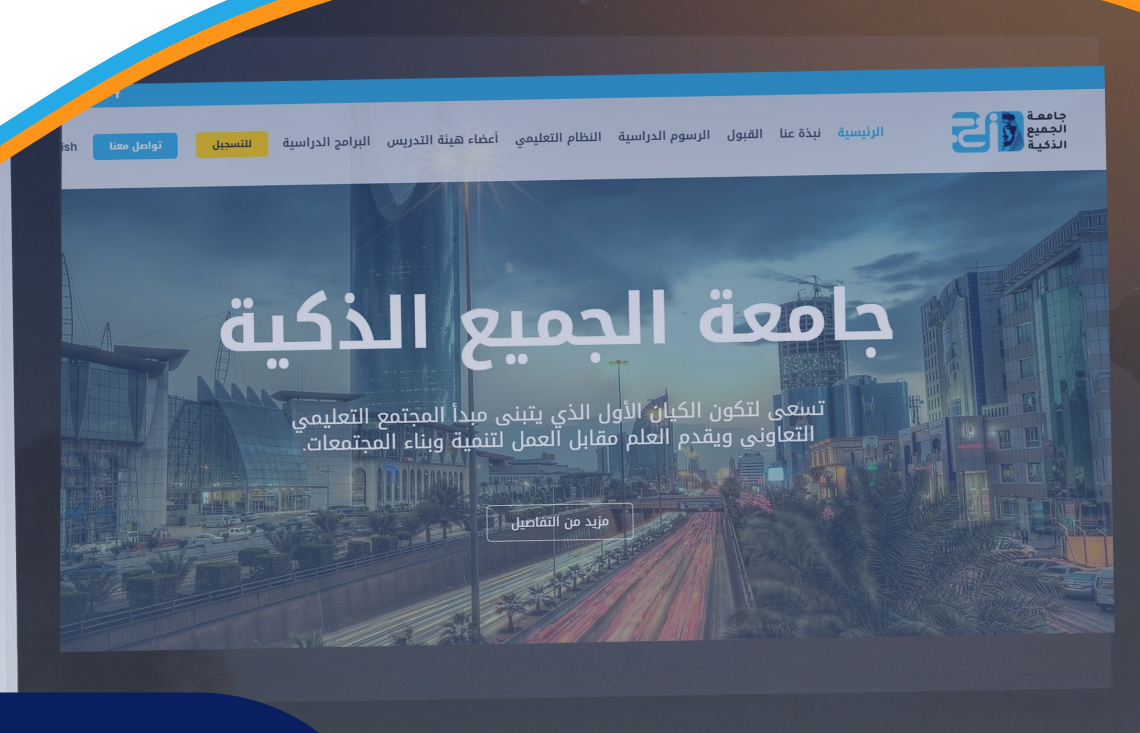




جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY



IoT Security
Prepared by Dr. Abdelrhman Ahmed
Lecture 9

Security Implementations of
IoT Systems

IoT authentication is a model for building trust in the identity of IoT machines and devices to protect data and control access when information travels via an unsecured network such as the Internet. Strong IoT authentication is needed so that connected IoT devices and machines can be trusted to protect against control commands from unauthorized users or devices. Authentication also helps prevent attackers from claiming to be IoT devices in the hope of accessing data on servers such as recorded conversations, images, and other potentially sensitive information.



There are several methods by which we can achieve strong authentication to secure IoT device communications:

- **One-way authentication:** in the case where two parties wish to communicate with each other, only one party will authenticate itself to the other, while the other party will not be authenticated.
- **Two-way authentication:** is also referred to as mutual authentication, in which both entities authenticate each other.
- **Three-way authentication:** is where the central authority authenticates the two parties and helps them to authenticate each other.
- **Distributed:** using a distributed straight authentication method between the parties to the communication.
- **Centralized:** using a centralized server or a trusted third party to distribute and manage the authentication certificates used.



IoT authentication types and suggested uses

One-way

- Remote devices transmitting low-value data.
- Organizations with few IoT devices located behind other security devices, such as firewalls or private networks.



Mutual or two-way

- Remote devices transmitting sensitive data such as financial information or healthcare data.
- Devices with a one-to-one relationship with the network or that only connect to single devices and servers.

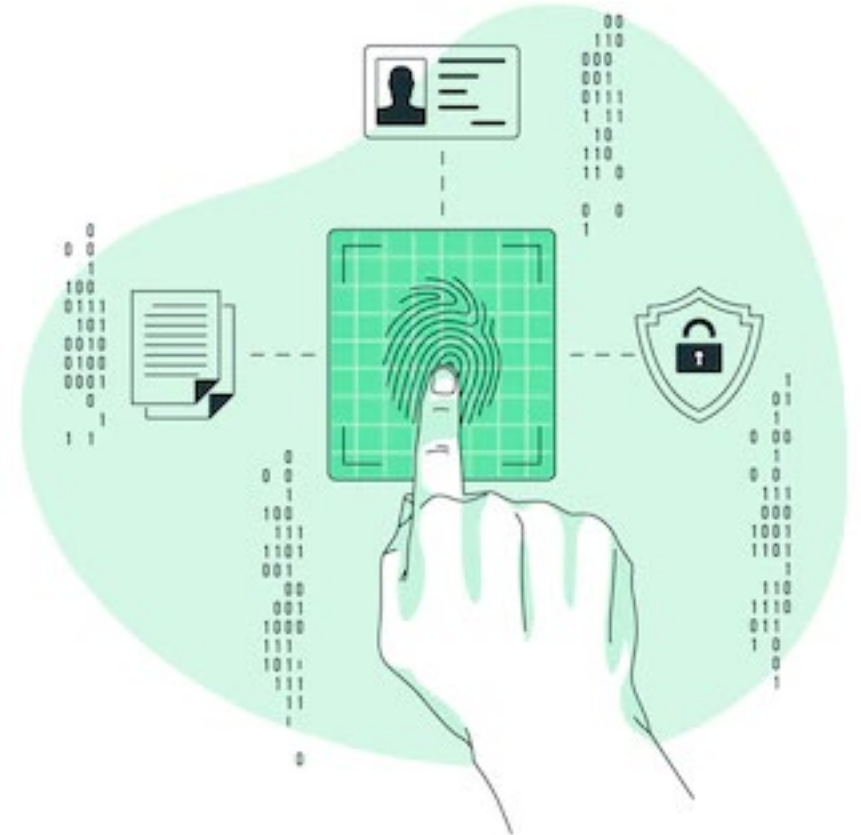


Three-way

- Organizations looking to manage a large fleet of IoT devices securely.
- Organizations that already have an infrastructure management solution monitoring their networks, systems and devices.



The Internet of Things (IoT) is not just a single technology, but a connected environment of various machines (“things”) that work together independently – without human interaction. The authorization process is the tool used to validate the identity of each endpoint in the IoT system. The certification process is configured upon enrollment entry and informs the service provider of the method to be used when checking the system’s identity during registration.



Machine Identity Management aims to build and manage confidence in a machine's identity that interacts with other devices, applications, clouds, and gateways.

This may include the authentication and authorization of IoT devices such as:

- Industrial Control Systems
- Connected medical devices
- Vehicle ECUs (Engine Control Units)
- Security cameras
- Home security systems
- Mobile devices
- Smart speakers, lights, outlets, and other equipment.



Each IoT machine needs a **unique digital identity** when connecting to a gateway or a central server to prevent malicious actors from gaining control of the system. This is accomplished through binding an identity to a cryptographic key, unique per IoT device.

- For trusted platform module (TPM) implementations, the registration ID is issued by the TPM itself.
- For X.509 certificates, the registration ID is issued by a globally trusted Certificate Authority (CA).

Machine identity management approaches are specifically responsible for discovering the credentials used by machines and the management of their life cycle.

Security falls into two main categories: **distributed and centralized**. In the distributed model, devices store certificates and identities and validate authorization. In the centralized model, a centralized server or trusted third-party application distributes and manages the IoT **devices' authentication certificates**. When devices connect to the network, the certificate's central repository performs the verification and authentication.

Depending on the nature of an organization's IoT devices, combinations of distributed and centralized models can ensure the most efficient and secure management. There are three main IoT authentication and authorization security protocols and options available that admins can deploy.

Distributed one-way authentication:-

Whenever two devices decide to connect, such as an IoT sensor and gateway, the protocol dictates that only one device authenticates itself to the other, and the second device is not authenticated. One device registers as valid with the second device through a password hash or digital certificate. When the first device tries to connect, the second device checks the password or certificate and compares it to the stored information. If the information matches, the device authorizes the connection.

One-way authentication works best for devices that only connect to one other device. These devices still need security mechanisms, but don't require constant monitoring.

Distributed two-way authentication:- Also known as mutual authentication, this protocol is used when both devices authenticate each other before they communicate. Each device must have a unique digital identity stored for the other device and then compare identities. The devices can only connect when the first device trusts the second device's digital certificate and vice versa. The Transport Layer Security protocol exchanges and compares certifications.

Online e-commerce transactions and highly sensitive data transmissions typically use this protocol.



Centralized three-way authentication:- In this approach, an admin registers the devices with a central authority or server and associates the devices with valid digital certificates. The central authority facilitates the secure handshake between the two devices that wish to communicate. In three-way authentication, the security certificates aren't stored on the devices and can't be stolen by criminals, yet the devices still have strong security. This approach works best for always-connected devices or ones with on-demand internet access because it eliminates any authentication delay. A **certificate and key lifecycle** management service can manage the certificates centrally and connect to any device on a network that needs verification.

Challenges Caused by Poor IoT Device Authentication

The rapid proliferation of IoT devices means that many networks are built on legacy authentication protocols, which are easier for hackers to circumvent. In addition, many newer IoT network architectures do not deploy best-in-class authentication security despite the risks of not doing so.

Simple certificates cannot address the multiple levels of authorizations, roles, and information these complex environments need.



Challenges Caused by Poor IoT Device Authentication

- **Identity and access management (IAM)**

IAM governs which devices connect and are allowed to do what within a network. It also defines how people or devices are identified and authenticated as authorized users. Many industrial IoT devices are located outside of the home network's security defenses, making them vulnerable to attack.

Once infiltrated, hackers can use them as a backdoor to steal important information on the server-side. Conversely, these devices are also vulnerable to fake messaging from spoofed servers, leading them to perform tasks they shouldn't. Without a completely secure IoT device authentication system, an organization's identity and access management function is never safe.

- **Data and network security**

IoT device networks create enormous amounts of data. This data is transmitted back to home networks, where it improves decision-making and system monitoring. There is also data sent towards the devices with instructions and updates. IoT device authentication is therefore critical for a number of reasons in this exchange.

Without secure authentication, there is no way to ensure the reliability of the data being received from IoTs or stop it from being tapped and stolen. With no way to authenticate their server, the devices can also install malicious software for use in other attacks.

Challenges Caused by Poor IoT Device Authentication

- **Regulatory compliance**

Multiple data protection regulations worldwide obligate organizations to strictly protect and manage any personally identifiable information they collect, including data received from IoT devices. However, industry-specific regulations, such as FDA requirements, also rely on secure IoT networks to ensure safety and compliance. Therefore, a network without strict IoT device authentication risks breaching regulations, incurring fines, and losing consumer trust.

- **Data sharing**

Securely analyzing and sharing data streams from IoT devices is key for successful collaboration with other organizations or third-party contractors. However, data sharing is becoming more difficult due to concerns over data safety. Secure and compliant sharing can only occur where clear IAM and IoT device authentication protocols are in place to reduce the risk of data breaches or oversharing.

IoT devices are often hacked remotely, involving a hacker trying to enter the device using an internet connection. **If an IoT device is only allowed to communicate with an authenticated server, any outside attempts to communicate will be ignored.** According to the 2018 Symantec threat report, the number of IoT attacks increased by 600 percent between 2016 and 2017, from 6,000 to 50,000 attacks, respectively. Therefore, when IoT devices are implemented within corporate networks, security needs to be given much more attention. To address this issue, powerful but efficient cryptography solutions must be used to standardize secure communication between machines. However, it is a tough decision to choose the right IoT authentication implementation model for the job. **Before deciding which architecture model is ultimately the best IoT authentication, you need to consider several factors, such as energy resources, hardware capacity, financial budgets, security expertise, security requirements, and connectivity.**

- **X.509 Certificates**

The X.509 protocol (IETF RFC 5280) provides the most secure digital identity authentication type and is based on the certificate chain of trust model. The use of X.509 certificates as a certification mechanism is an excellent way to scale up production and simplify equipment delivery.

Public key infrastructure (PKI) consists of a tree-like structure of servers and devices that maintain a list of trusted root certificates. Each certificate contains the device's public key and is signed with the CA private key. A unique "thumbprint" provides a unique identity that can be validated by running a crypto algorithm, such as RSA.

- **X.509 Certificates**

Digital certificates are typically arranged in a chain of certificates in which each certificate is signed by the private key of another trusted certificate, and the chain must return to a globally trusted root certificate. This arrangement establishes a delegated chain of trust from the trusted root certificate authority (CA) to the final entity “leaf” certificate installed on the device through each intermediate CA. **It requires a lot of management control, but there are many vendor options out there.**

However, X.509 certificate lifecycle management can be a challenge due to the logistical complexities involved and comes at a price, adding to the overall solution cost. For this reason, many customers rely on external vendors for certificates and lifecycle automation.

- **Hardware Security Module (HSM)**

The Hardware Security Module, or HSM, is used for secure, hardware-based device secret storage and is the safest form of secret storage. Both the X.509 certificate and the SAS token can be stored in the HSM. HSMs may be used with the two attestation mechanisms supported by the provisioning service.

Alternatively, device secrets may also be stored in software (memory) but is a less secure form of storage compared to an HSM.

Online Course

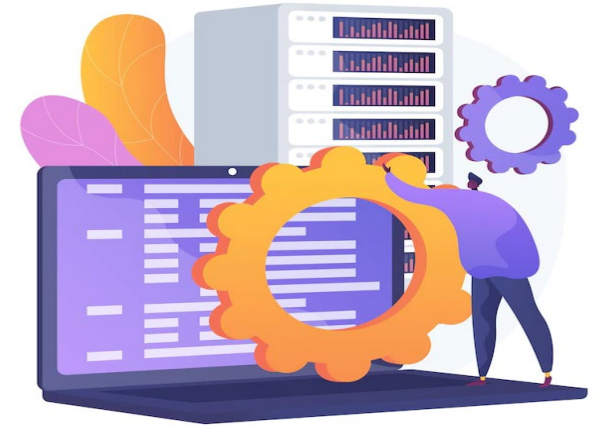


- **Trusted Platform (TPM) Module**

It is essential to check the device's identity that communicates with the messaging gateway in IoT authentication deployments. The usual method is to generate key pairs for devices that are then used to authenticate and encrypt traffic. However, the disk-based key pairs are susceptible to tampering.

TPMs come in a number of different forms, including:

- Discreet hardware devices
- Embedded hardware equipment
- Implementation of firmware
- Implementation of software



- **Trusted Platform (TPM) Module**

While a typical TPM has several cryptographic capabilities, **three key features are relevant to IoT authentication:**

- Secure boot-up
- Establishing the root of trust (RoT)
- Identification of device

Device manufacturers cannot always have full confidence in all entities in their supply chain (for example, offshore assembly plants). Still, they cannot give up the economic benefits of using low-cost suppliers and facilities. The TPM can be used at various points along the supply chain to verify that the device has not been incorrectly modified.

- **Trusted Platform (TPM) Module**

The TPM has the capability to store the keys securely in the tamper-resistant hardware. The keys are generated within the TPM itself and are therefore protected from being retrieved by external programs. Even without harnessing the capabilities of a trusted hardware root and a secure boot, the TPM is just as valuable as a hardware key store. Private keys are protected by hardware and offer much better protection than a software key.

With TPM, you can't roll the key without destroying the identity of the chip and giving it a new one. It's like if you had a clone, your clone would have the same physical characteristics as you, but they're a different person in the end. Although the physical chip remains the same, your IoT solution has a new identity.

- **Trusted Platform (TPM) Module**

Some key differences between TPMs and symmetric keys (discussed further below) are as follows:

1. TPM chips can store X.509 certificates as well.
2. The TPM certificate in the Device Provisioning Service uses the TPM endorsement key (EK) which is a form of asymmetric authentication, while the symmetric keys are symmetric authentication.
3. TPM certification is more secure than SAS token-based symmetric key attestation.
4. Difficult to develop without either a physical TPM or a quality emulator.
5. May require a redesign of the board to be included in hardware.
6. New IoT devices should be fundamentally designed to support a TPM.

- **Symmetric Keys**

Symmetric Key Certification is a simple approach to authenticating a device with a Device Provisioning Service instance. This certification method is the “Hello World” experience for developers who are new to or do not have strict safety requirements. Device attestation using a TPM or an X.509 certificate is more secure and should be used for more stringent safety requirements.

Symmetric key enrollments also provide a great way for legacy devices with limited security features to boot into the cloud via Azure IoT.

The symmetric key attestation with the Device Provisioning Service is carried out using the same security tokens supported by IoT hubs to identify the devices. These security tokens are SAS (Shared Access Signature) tokens.

- **Symmetric Keys**

SAS tokens have a hashed signature created using a symmetric key. The signature shall be recreated by the Device Provisioning Service to verify whether or not the security token presented during the certification is authentic.

When the device certifies with an individual enrollment, the device uses the symmetric key defined in the individual enrollment entry to create a hashed signature for the SAS token.

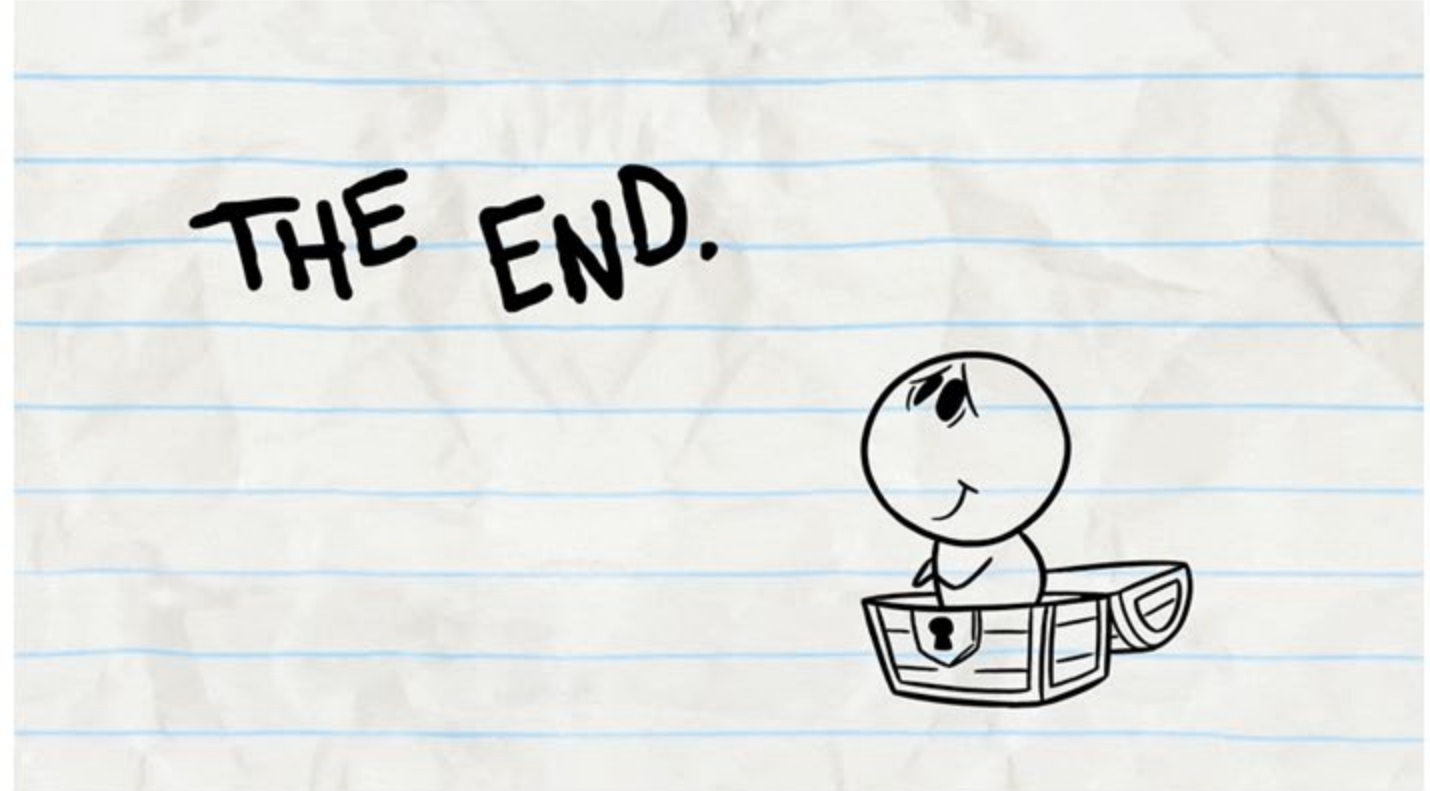
Shared symmetric keys may be less secure than X.509 or TPM certificates because the same key is shared between the device and the cloud, which means that the key needs to be protected in two places. Designers using symmetric keys sometimes hardcode the clear (unencrypted) keys on the device, leaving the keys vulnerable, which is not a recommended practice



The End

Please feel free to contact me anytime if you have any questions

Abdelrhman Ahmed Refaat:
abdelrhmanahmedrefaat@gmail.com



جامعة القادسية
EVERYONE'S SMART UNIVERSITY