Digital twins for industrial IoT with Altior

A technical white paper

Part 3: Aegis security

InkwellData
Connected Innovation

Digital twin for industrial IoT with Altior: Aegis security

Introduction

In the Altior IoT service platform, security is a building block of the overall design. This means that Altior components include security as a core element.

Altior is a cloud-based platform with a focus on industrial IoT (IIoT) support, with the aim of providing data transport and manipulation functions to field devices and sensors.

In this document we introduce the basic concepts of Altior data and information security at the device level, and its integration to the infrastructure and the data management level.

Note: for Inkwell Data, data security is not simply based on features and products, but data security is the product of a process. Inkwell Data doesn't rely on specific implementation of proprietary algorithms.

On the contrary, all the security implementation features are based on well documented and where possible open-sourced technology, to give total accountability of the technology used to customers and end users.

General IoT security considerations

Each IoT device provides one or more capabilities—features or functions—it can use on its own or in conjunction with other IoT and non-IoT devices to achieve one or more goals.

Some of the most relevant features for Altior security are the following:

- **Transducer** capabilities interact with the physical world and serve as the edge between digital and physical environments. Transducer capabilities provide the ability for computing devices to interact directly with physical entities of interest. Every IoT device has at least one transducer capability. The two types of transducer capabilities are:
    - Sensing: the ability to provide an observation of an aspect of the physical world in the form of measurement data. An example is temperature measurement.
    - Actuating: the ability to change something in the physical world. An example of actuating capability is a remote valve shut off.
- **Data** capabilities are typical digital computing functions involving data: data storing and data processing.
- **Interface** capabilities enable device interactions (e.g., device-to-device communications, human-to-device communications). The types of interface capabilities are:
    - **Application interface**: the ability for other computing devices to communicate with an IoT device through an IoT device application. An example of an application interface capability is an application programming interface (API).
    - **Human user interface**: the ability for an IoT device and people to communicate directly with each other. An example of human user interface capability is a display.

- **Network interface**: the ability to interface with a communication network for the purpose of communicating data to or from an IoT device—in other words, to use a communication network. A network interface capability includes both hardware and software (e.g., a network interface card and the software implementation of the networking protocol that uses the card). Examples of network interface capabilities includes Ethernet, Wi-Fi, Bluetooth and Weightless. Every IoT device has at least one enabled network interface capability and may have more than one.
- **Supporting** capabilities provide functionality that supports the other IoT capabilities. Examples are device management, cybersecurity, and privacy capabilities.

Security goals for Altior

An IIoT network shares some commonalities with a computer network, but also poses unique requirements; thus, we combine the security objectives and requirements of any computer system with those peculiar of field deployed sensors and devices.

Confidentiality

Confidentiality is the ability for the system to keep data confidential and prevent inadvertent disclosure. Confidentiality ensures the concealment of the message from an attacker so that any message communicated via the sensor network remains confidential.

In a IIoT network, the issue of confidentiality should address the following requirements: (i) a sensor node should not allow its readings to be accessed by its neighbours unless they are authorized to do so, (ii) key distribution mechanism should be extremely robust, (iii) public information such as sensor identities, and security keys of the nodes should also be encrypted in certain cases to protect against traffic analysis attacks.

Integrity

Integrity means that the data must be accurate, for example the temperature readings should reflect the real temperature. Integrity ensures the reliability of the data and refers to the ability to confirm that a message has not been tampered with, altered or changed while on the network. In a IIoT network, the issue of integrity should address the following requirements: (i) only the nodes in the network should have access to the keys and only an assigned base station should have the privilege to change the keys. This would effectively thwart unauthorized nodes from obtaining knowledge about the keys used and preclude updates from external sources. (ii) It protects against an active, intelligent attacker who might attempt to disguise his attack as noise.

Availability

Availability means data must be available when needed and relevant. Availability ensures the services of resources offered by the network, or by a single sensor node must be available whenever required. In an IIoT network, the issue of availability should address the following requirements: (i) the security mechanisms should be available all the time; a single point of failure should be avoided, (ii) the mechanism is used as a central access control system to ensure successful delivery of every message to its recipient node.

Authentication

Authentication ensures the reliability of the message by identifying its origin. By authenticating other nodes, cluster heads (when present), and base stations before granting a limited resource, or revealing information. In a IIoT network, the issue of authentication should address the following requirements: (i) communicating node is the one that it claims to be, (ii) receiver node or base station should verify that the received packets have undeniably come from the actual sender node.

Data freshness

It is also needed that the data is recent and no old message have been replayed. To achieve this goal a nonce, or another time related counter is added to the packet.

Time synchronization

Most of the IIoT network applications depend on some form of time synchronization. Sensor may also require to compute end-to-end delay of a packet as it travels between itself and its base station.

Secure Localization

The utility of an IIoT network also relies on its ability that it automatically and accurately locates each sensor in the network. In order to accurately point out a failure a designed IIoT network will need accurate location information. An attacker can easily exploit this situation and can easily manipulate non-secured location information by either reporting false signal strengths or replaying signals.

Aegis: The Altior security framework

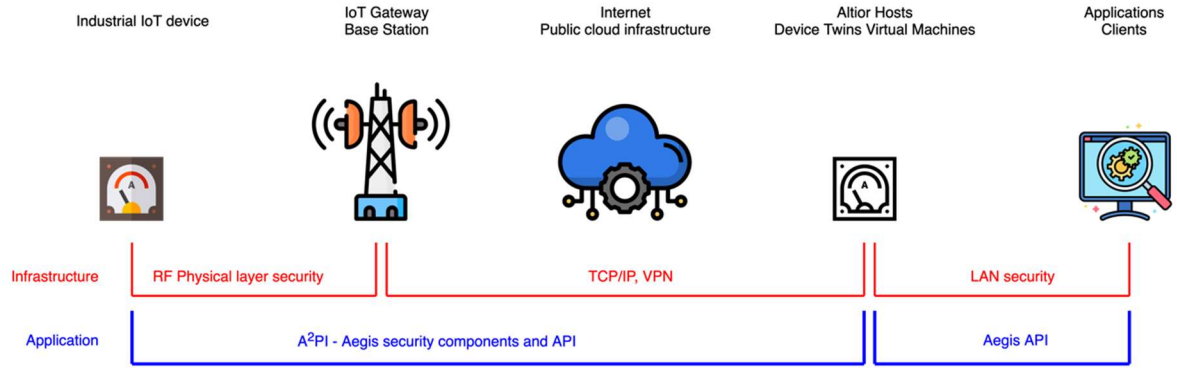
To address the security requirements of IIoT services, Altior implements a security framework called **Aegis** and a corresponding API called **Aegis API**, which are based on the Altior API (Application Programming Interface), A²PI.

The A²PI is an open API implemented in different components of the Altior platform to ensure seamless integration of sensors/devices and IT systems.[1]

The Aegis security API is built on the assumption that any segment of a communication network can be non-secure can be vulnerable to intended or unintended attacks; however, the security A²PI makes no assumptions on the communication network technology or network topology as the API relies on the abstraction provided by the network middleware features implemented by Altior (more about this on the following sections).

End to End security

Aegis implements a layered end-to-end security model, where all the data traffic is secured by a common encryption system that covers the physical and data transport layers as well the application layer.



Note that Aegis is designed to provide primarily security and protection for the data in transit or data in motion.

By "data in transit" we define data that is actively moving from one location to another such as across the internet or through a private network. Data protection in transit is the protection of this data while it's traveling from network to network or being transferred from a destination storage[2].

The counterpart of the data in transit is data at rest, that is data that is ultimately stored on a physical device (hard-disk drive, SSD drive, flash drive, etc.). This is data that has a different risk profile from data in transit and it is subject to different attacks.

---

[1] For more information about the Altior open A²PI, please refer to part 5 of the Altior architecture white papers.

[2] Using this definition of data in motion, the data managed by the Altior digital device twins is also considered in motion, as the data handling by device twins is based on the ephemeral lifecycle of device twins' instances managed by the Altior virtual machine. For more information about this, please see next section.

Some Aegis features like the Aegis-KM secret storage manager can be accessed by third party applications and used to provide security to data at rest, although this should be considered just a helper tool in the company's data protection strategy.

Infrastructure Level Security

At the infrastructure level (devices and data transport), Aegis provides a mechanism to securely connect the sensor/device to the network backend through the base station.

The A^2PI embedded implementation is designed to leverage on the secure features of modern industrial microcontrollers. The API has a public (untrusted) layer that provides all the generic connection setup and control relative to the actual LPWAN connectivity layer used.

The authentication and encryption functions are based on the underlying hardware level for basic elements such as the actual crypto computations, to save space on the circuit board and conserve energy. The API device side models a virtual secure machine, and its implementation relies on hardware security for the following main areas:

Authenticity	Unique and immutable identity Cloning and counterfeiting prevention Certificates protection
Data confidentiality	Keys protection Credential protection Local data protection
Firmware integrity	Code integrity Secure vs insecure isolation Secure communications
Device integrity	Tamper prevention Genuine verification

The basic functions provided by the secure hardware platform include (but are not limited to):

- AES crypto engine NIST FIPS 197 implementation, supporting up to 256 bits key size, key derivation and key decryption, and the basic chaining modes: ECB, CBC, CTR, GCM, GMAC and CMAC;
- Hash accelerator processor in compliance with FIPS 180-2, supporting secure hash standards SHA-1*, SHA-224, SHA-256 and the IETF RFC 1321 MD5*;
- True random number generator based on a noise source and equipped with a fault detector.

The Aegis API is provided to the developers and its use is mandatory, so no Altior-approved device can be deployed without a basic secure layer in place.

To use the secure API, the firmware developers only need to implement basic call back functions in their proprietary code. This also has the side effect of standardising and optimizing the energy usage of the security operations for battery powered devices, making battery life predictions easier for the application designer.

However, for advanced developers or specific project requirements, the API offers direct access to the hardware implementation of basic security features through a common set of function signatures, that makes the development more portable from one hardware platform to another. Direct access can also be used to augment the existing features or to implement a custom security layer that can be replicated on the network backend.

For LPWAN-based networks where IoT data gateways are used to cover the "last mile" of the sensor connectivity, Inkwell Data offers an Aegis implementation for the gateways with the Inkwell Data network gateway (NG) design. The NG is a virtual network edge device designed to host one or more LPWAN-U[3] based network interfaces.

Since the Aegis security API is implemented on Inkwell Data's network gateway[4], the default security model allows for secure identification and authentication of every node added to the network. This doesn't prevent the application developers to implement their own additional security layer operating at the base station level as well.

On public LPWAN networks (like NB-IoT and LTE- Cat M), the Aegis API relies on the implementation of the default security provided by the network operator. Although public networks are deemed intrinsically secure, Aegis allows further wrapping of the data in motion based on the data stream managed by the Altior network middleware[5] (see section "Extending Aegis").

[3] LPWAN-U is a Low Power Wide Area Network technology operating on unlicensed radio frequency spectrum. Examples of LPWAN-U supported by Altior are LoRa and LoRaWAN, wireless MBUS mode N2, Sigfox, Weightless.

[4] The Inkwell Data IoT network gateway specification is available to Inkwell Data partners as an open-source reference design. For more information about Inkwell Data Altior Ecosystem Partner Programme, please contact aepp@inkwelldata.com

[5] For more information about the Altor networking middleware architecture see: https://inkwelldata.com/wp-content/uploads/2021/03/Digital-twins-for-industrial-IoT-with-Altior-Part-2.pdf

Application-Level Security

In the Altior cloud/data centre, the API own security is implemented with the same feature set available at the infrastructure level.

This means that in standard operations, all the uplink and downlink data traffic is secured in a straightforward manner since the relevant API levels do the identification, authentication, encryption and decryption using the same function set.

As long as the correct credentials are provided, all the security processing is done by specific processes modelled after the device they are operating on, that in Altior are called device digital twins.

That's because inside the Altior platform, every device is maintained as a separate and isolate computational process (the digital device twin), becoming active only when there is actual data traffic related to the specific object instance[6].

Simplifying, an Altior device digital twin (DT) is a virtual clone of the real device, running in the Altior cloud; the DT is a reactive object that operates on the data received or transmitted on the network infrastructure through the filtering of the security subsystem.

The Zero Trust approach for device twins

Altior Aegis provides a zero-trust security model. Zero trust at its core assumes that no user is trusted (whether an internal or external user).

Users[7] of all types are considered a possible threat, and the actions those users can perform and the resources the users can access on a network or on a system should be limited by default.

One of the concepts of the zero-trust network architecture is "least privileged access".

Least privileged access involves restricting access down to the most granular level, meaning users should have access only to the systems, servers or applications needed to do their job and nothing more.

For human users, least privileged access is accomplished using Privileged Access Management (PAM) or Identity Access Management (IAM) systems that help manage each user's access permissions.

Aegis implements Zero Trust for device twins with a combination of access control methods named Risk Adaptive Access Control (RADAC). RADAC incorporate the benefits of ACL (Access Control List), RBAC (Role Base Access Control) and ABAC (Attribute Based Access Control).

Every device twin instance has a RADAC profile that is used to check for legitimate access to the Aegis data infrastructure, and every single operation performed by the device twins, Altior applications and API invocation is checked against their RADAC profile.

Since the RADAC check is a core element of the Altior device twins' virtual machine, checks are computationally inexpensive and barely noticeable even under a heavy traffic load.

---

[6] For more information about digital device twins, see the Altior digital twin whitepaper: https://inkwelldata.com/wp-content/uploads/2021/03/Digital-twins-for-Industrial-IoT-with-Altior-Part-1.pdf

[7] In the Aegis context, users are not just human users, but also internal and external software components. A user is any entity that can operate on an Altior device twin.

Extending Aegis

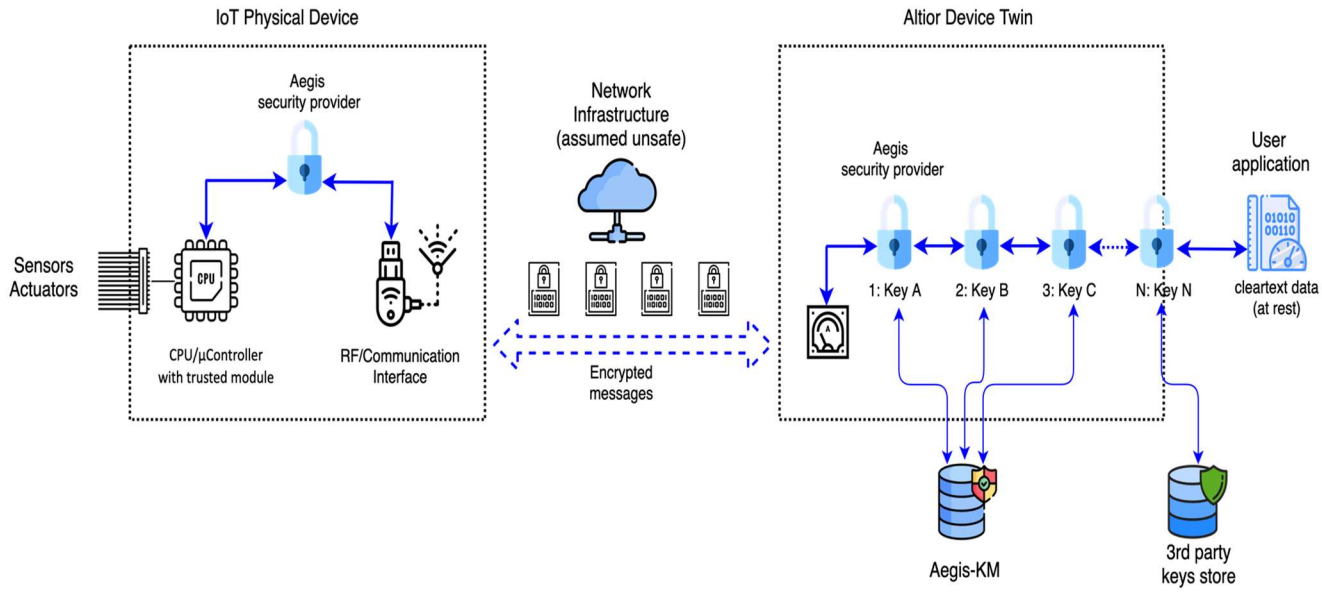
As mentioned in the previous section, the default Aegis security behaviour can be enhanced with the custom implementation of new security features defined in the device itself and in the management platform by using the A²PI extension capabilities.

The Altior device twin virtual machine implements a dataflow-based application paradigm that allows the inclusion of plug-in modules in the flow processing cycle.

The security plug-in modules are called “security providers”, and are included in the data process flow of the digital twin.

Security providers for a single digital twin are called in an ordered sequence (a call chain) that is transactional in nature: if a single step fails, the whole operation is aborted.

Each security provider has a different encryption key(s) and they can get their keys from external datastores.



The security provider mechanism makes straightforward the implementation of one or more additional security layer on top of the default one, or in extreme cases totally replacing the Altior security features.

A security providers chain can be imagined as a series of encrypted envelopes where the encryption keys can be managed by different users or agents. This allows a further segmentation of the data access, with the added bonus that different security providers can operate on different segments of the data message, so that for example only selected portions of the cleartext of a message can be provided to a specific user while keeping the rest of the data encrypted.

Security providers can be developed by network operators or even by end users, and are tested by running inside “sandbox” processes inside the Altior platform.

The sandbox model makes also easier to debug the firmware security provider implementation on the physical device, providing a safe and secure testing environment.

Cryptography keys and secrets management

Altior security infrastructure is built on industry tested, open building blocks.

Altior basic security components such as cryptography algorithms are open source, using OSI approved licenses, because they are built on existing open-source components considered industry standards.

This allows the Altior team to leverage on the continuous security audit and security patching done by the open-source community; on the other hand, Inkwell Data is a serious contributor to the open-source world making new implementations available for the general public.

In a cybersecurity context, “open” means also that the security features of the Altior platform are published and subject to the scrutiny of all the interested parties: from developers to security consultants.

It is through this positive feedback loop that the Altior platform will maintain an industry grade security level as well as improve its security features.

Aegis-KM

Given that the Altior security procedures are in the open, the strength of the single application lies on the quality and the management of the encryption keys.

Managing encryption keys is a challenging task by itself: because of the massive scale of IIoT objects roll out, the number of keys to maintain can be staggering.

Moreover, encryption keys in IIoT may have a definite life-cycle and updating and keeping track of key changes is a daunting task.

For these reasons, Altior implements an advanced key management server: Aegis-KM.

Aegis-KM is a FIPS 140-2 compliant enterprise key manager server for securely managing keys and generic secrets and encrypting data in-transit.

From storing credentials and API keys to encrypting passwords for user signups, Aegis-KM is the ideal secure repository.

Aegis-KM is a symmetric encryption key management solution that creates, manages, and distributes 128-bit, 192-bit, and 256-bit AES keys for any application and it is specifically tailored for the requirements of IIoT.

For sensitive applications, Aegis-KM encryption and key management can help to meet the privacy requirements of the EU General Data Protection Regulation (GDPR).

The main Aegis-KM features include (but are not limited to):

- **Secure secret storage**: Arbitrary key/value secrets (tuples) can be stored in Aegis-KM. These tuples are encrypted prior to being written to persistent storage.
- **Dynamic secrets**: Aegis-KM can generate secrets on-demand. After creating these dynamic secrets, Aegis-KM will also automatically revoke them after the lease is up.
- **Leasing and renewal**: each key in Aegis-KM has a lease associated to it. At the end of the lease, Aegis-KM will automatically revoke that key.

- **Revocation**: under certain circumstances, keys should be revoked, and Aegis-KM has built-in support for revocation.
- **On-the-fly data encryption**: Aegis-KM can invoke Aegis crypto services to encrypt and decrypt data without storing it. The encryption key never leaves the key server device with on-the-fly encryption services, and this feature can be used for higher level security applications such as payment processing.
- **Auditing**: Aegis-KM built in logging allows administrators to track all key retrieval, key management, and system activity. Reports can be sent automatically to central log management, alerting facilities, or SIEM products for a timely and permanent record of activity.
- **Key Change and Rotation**: Aegis-KM automatically or manually rotate encryption keys. Security administrators can define the frequency of key rotation based on internal security policies. When a key change occurs, the new version is created and the old version is moved to a historical database and available for cryptographic operations.

Aegis-KM is designed for central key management with distributed execution. This architecture enables the encryption and decryption node to exist at any point within the Altior network.

The Aegis-KM key management component can easily be deployed onto different nodes and can be integrated with any encryption application, with the flexible Altior plug-in system.

Once deployed, all the encryption/decryption mechanisms are available at the node level, where the encryption/decryption task is performed. This approach reduces the data's network trips.

This approach also reduces the risk of application downtime because of the failure of a single central component. The key manager is responsible for managing the generation, secure storage and expiration of the keys which are used by all the platform clients.

Key distribution for IoT sensors and devices

Sensor nodes are limited in terms of computing, memory and energy capacities. These limitations affect negatively the functioning of the special smart techniques that provide the required security. For energy saving, and to keep computational complexity reasonably low, the Altior platform employs a symmetric cryptography scheme.

In symmetric cryptographic techniques, a single shared key is used between the two communicating nodes both for encryption and decryption. This shared key has to be kept secret in the network, which can be quite hard in the exposed environment where sensors are used.

Most security schemes for IIoT use only symmetric cryptography, due to its ease of implementation on limited hardware and small energy demands, especially if the implementation is done in hardware to minimize performance loss.

When sensor networks are deployed in a hostile territory or a special region, they should secure the communication between two sensor nodes by encryption/decryption, safety authentication techniques and others.

For these reasons, Aegis-KM is optimized for IIoT symmetric keys distribution using advanced techniques, and the A^2PI includes the equivalent technology on the device side.

Key management protocols provide safe paths in IIoT networks, and Altior employs state of the art key distribution protocols. The A^2PI in fact, implements a few key distribution clients, including (at the time of writing) a time-based enhanced LEAP (Localized Encryption and Authentication Protocol), an optimized IKEv2 (Internet Key Exchange) protocol implementation, and provides support for Aegis-based key pre-distribution scheme based on µ-PBIBD design.

All the key distribution protocols implemented in Aegis are tested against the most known attacks like spoofed, altered, or replayed routing information, selective forwarding and the Sybil attack, sinkhole and wormhole attacks, jamming attacks (DoS) like node subversion and malfunction, physical attacks like false node and node replication attacks and passive information gathering.

Conclusions

There is no such thing as a one-size-fits-all approach to security, and each framework has its pros and cons. The Altior Aegis security framework is designed with the requirements of the IIoT network operator in mind, but it is open to any kind of integration due to its flexibility and versatility.

Aegis is not a silver bullet product to solve all the security issues when implementing an IoT based service, rather Aegis components and APIs should be considered a support to build a secure data management process.

Implementing Aegis and Altior does not exempt the enterprise to apply common sense and best practices of the information security industry. Implementing robust network security by means of network firewalls and VPNs, web application firewalls and regular, periodic vulnerability scans are still required to achieve some confidence in the overall data security infrastructure.

Aegis uniqueness lies in its implementation at multiple levels in the IIoT value chain, allowing a shorter time to market to industrial IoT application, while its openness can be a key value for an expanding ecosystem.

The Inkwell Data research and development team is focused on keeping Aegis constantly updated to guarantee a continuous service to the Inkwell Data customers.

For more information and contact requests:

Inkwell Data ltd
Iveagh Court
Harcourt Road
Dublin 2 - Ireland
+44 (0)203 951 0831

info@inkwelldata.com

United House
9, Pembridge Road, Notting Hill
W11 3JY London - UK
+44 (0)203 951 0831

https://inkwelldata.com

Inkwell Data Italia
via degli Zabarella, 24
35121 Padova - Italy
+39 (0)49 689 8423

Cover photo by Jonathon Young on unsplash.com