Data Protection Best Practices

An Industrial Internet Consortium White Paper

Version 1.0

2019-07-15

Protecting sensitive data created, stored and consumed by sensor-driven Industrial Internet of Things (IIoT) technology and applications is one of the foundations of trustworthy IIoT systems.

Broadly speaking, data protection measures resist internal and external disturbances and attacks on critical data and IIoT systems at large. These measures are applied over the entire lifecycle of the data from when the data is generated to when the data is destroyed or securely archived. Applying appropriate measures to data-at-rest, data-in-motion, and data-in-use, promotes confidence in the security of the broader IIoT system.

Failure to apply proper data protection measures can lead to serious consequences for IIoT systems, such as:

- service disruptions that in turn affect the bottom-line,
- serious industrial accidents and
- significant regulatory fines, loss of IP, and negative impact on brand reputation, caused by data leaks, as well as the failure to detect them and report them in a timely fashion.

This whitepaper describes measures needed to achieve a desired level of security for data, as well as safety, reliability, resilience, and privacy, so that the reader can more easily apply existing best practices. It is based on detailed analysis in existing industrial guidance and compliance frameworks such as IISF [IIC-IISF2016], IEC 62443, IEC 61508, and CSA CCM, which cover various aspects of industrial internet security and various aspects of data protection.

This document covers the best practices for various domains and sub-domains under the umbrella term data protection. It complements the IoT Security Maturity Model [IISM)[1] and builds on the concepts of the Industrial Internet Reference Architecture [IIRA[2]] and Industrial Internet Security Framework [IISF[3]]. The IIC Vocabulary Technical Report[4] provides terminology and definitions for this and other IIC documents. This document does not cover related aspects of equipment safety, certificate management, nor communications.

Existing and subsequent and Industrial Internet Consortium (IIC) Best Practices documents cover other aspects of industrial internet security, based on the six building blocks of the IISF.

In this paper we use “IoT” to refer to *all* Internet of Things systems, and “IIoT” when we wish to restrict our area of concern to Industrial IoT (IIoT) systems that connect and integrate industrial control systems with enterprise systems, business processes, and analytics [IIC Vocabulary].

---

1 https://www.iiconsortium.org/pdf/SMM_Description_and_Intended_Use_2018-04-09.pdf

2 https://www.iiconsortium.org/IIRA.htm

3 http://iiconsortium.org/IISF.htm

4 http://iiconsortium.org/vocab/index.htm

SOME DEFINITIONS AND CONTEXT

Here are some definitions of terms from organizations like the IIC, OMG and IEC:

Data protection is an umbrella term that covers adjacent and overlapping domains: data security, data integrity, and data privacy. Some security experts use 'data protection' interchangeably with 'data security', but this paper extends data protection to cover other aspects, including integrity and privacy.

The figure below illustrates these assumptions and assertions:

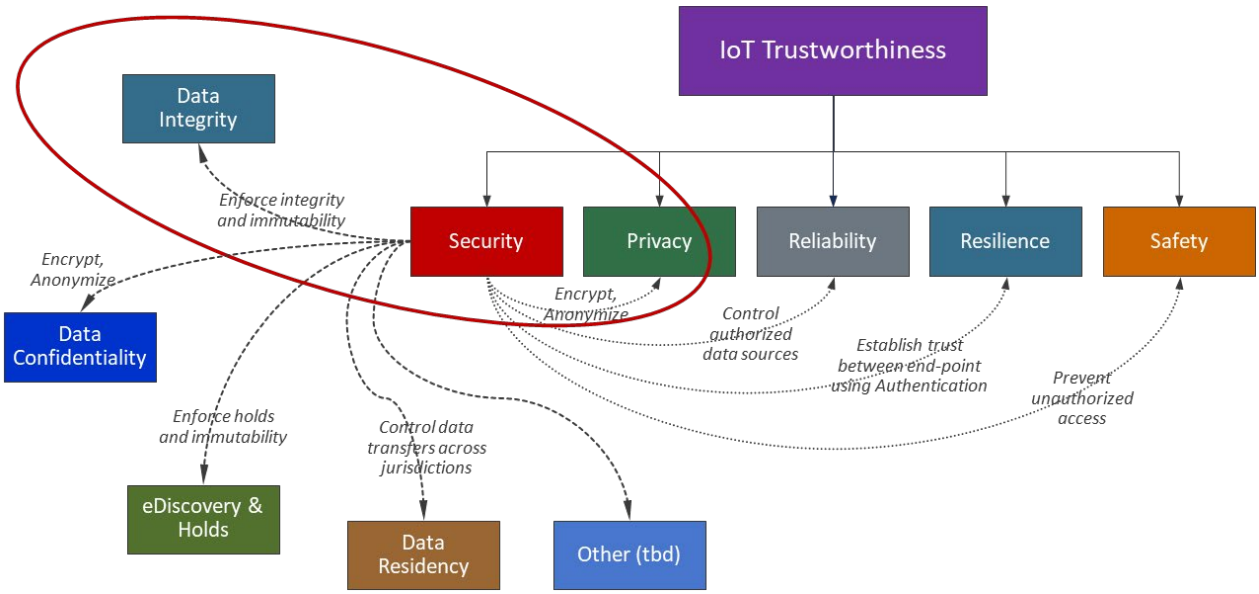

Figure 1—Data protection and the important role of Data security [source IGnPower]

Data security: Property of being protected from unintended or unauthorized access, change or destruction ensuring availability, integrity and confidentiality [IIC Vocabulary].

Data integrity: Property that data has not been altered or destroyed in an unauthorized manner.[5]

Data privacy: Right of individuals to control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed.[6]

[5] ISO/IEC 27040:2015: Information technology—Security technique—Storage security, 2015 January, https://www.iso.org/standard/44404.html

[6] ISO/TS 17574:2009: Electronic fee collection—Guidelines for security protection profiles, 2009 September, https://www.iso.org/standard/52387.html

Another term used in the industry and relates to data protection is *data confidentiality,*[7] which is a property that information is not made available or disclosed to unauthorized individuals, entity or processes [IIC Vocabulary].

We focus on the best practices for these data protection domains, and emphasize the core and enabling role that data security plays in them.

Acronyms and additional terms relevant to this model are defined in the appendices.

Data Breaches can lead to Multiple Negative Consequences

Data breaches can violate multiple data protection requirements, which in turn can lead to multiple negative consequences.

For example, a data breach that affects personal data is a violation of data privacy laws. If that data is business-sensitive, then that same breach also violates data confidentiality requirements.

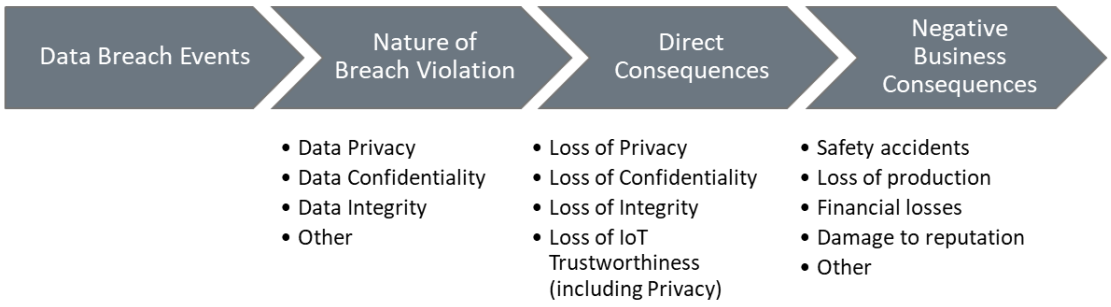

Figure 2—Data breaches can lead to multiple consequences [source IGnPower]

Data protection strategies must prevent and mitigate multiple types of data protection risks. This requires active engagement and coordination by the data security resources with the functional areas affected by the resulting negative consequences.

Categories of Data To Be Protected

Data protection touches all data and information in the organization, for example:

- operational data,
- system and configuration data,
- personal data and
- audit data.

[7] ISO/IEC 27000:2016

Operational data: In IoT, operational data refers to any data produced at the field site during the normal business operations. This data is generated either by sensors placed in the field or by electronic equipment and controllers like SCADA[8] equipment, ICS[9] controllers, or network equipment. Operational data includes data generated by sensors attached to physical processes and field equipment, data generated by electronic controllers and devices, and control data sent to the field environment from external networks.

This data has two main uses: deriving process insights for condition-based maintenance and performing monitoring and control of field equipment.

System & configuration data refers to data exchanged with an IoT device to enable it to function and operate in accordance with the design and operational requirements. This data must be protected in motion and at rest.

Personal data[10] is data (or information) that can identify individuals or specific characteristics about individuals, such as:

- name, gender, national ID, location, date of birth, home address, cultural, social characteristics, online computer identifiers,
- physical, genetic, psychological, mental, medical, financial,
- recruitment, salary, performance, benefits and
- race, ethnicity, religion, political opinions, biometric.

Privacy laws and regulations[11] impose restrictions on the handling of personal data, and the definition varies by jurisdiction.

Audit data is the "chronological record of system activities to enable the reconstruction and examination of the sequence of events and changes in an event*",* according to the US NIST Glossary of Key Information Security Terms.[12] IoT systems produce audit data at various levels of the architecture layers (edge, cloud, etc.) It must be protected to ensure its authenticity and reliability.

Data-at-Rest vs Data-in-Motion vs Data-in-Use

The IISF breaks down the elements for data protection into several categories:

- endpoint data protection,
- communications data protection,

---

8 Supervisory Control And Data Acquisition

9 Incident Control Systems

10 The definition of "personal data" varies according to laws, regulations and jurisdictions.

11 Example: US HIPAA, US Federal Privacy Act, EU GDPR, California CCPA, Canada PIPEDA, etc.

12 NIST Information Security Glossary https://nvlpubs.nist.gov/nistpubs/ir/2013/NIST.IR.7298r2.pdf

- configuration data protection and
- monitoring data protection.

Different data protection mechanisms and approaches are used depending on whether the data is at rest (DAR), in use (DIU), or in motion (DIM). Below are definitions of these terms (more background in the IIC IISF document).

Data at rest refers to data stored at various times during its lifecycle. Data at rest is vulnerable to manipulation and its confidentiality, integrity, availability (CIA) must be protected. Data encryption and replication are the most common techniques used to ensure CIA protection for data at rest. AES-XTS[13] (originally called Bitlocker[14]) is used extensively in enciphering fixed length sectors on non-volatile (removable) storage media. AES-XTS was specifically designed to protect and defend data at rest against attacks unique to this space. It is not recommended for other applications like data in motion (and algorithms used to protect data in motion are also not suitable for use in the data at rest context).

Data in motion refers to the data being shared or transmitted from one location to another. Networks being one of the most vulnerable points in a system, the data should be protected while it is motion. Network level security using TLS[15] is the most common method to protect data in motion. Since TLS is point-to-point, the endpoints of the TLS channel must be trusted and intermediate links avoided.

Data in use refers to the data that is being processed. When data is in use, if it is sent from memory to the processor unencrypted, it could be vulnerable to attacks. Data transformation, access control and secure memory are some of the ways to protect data in use, see “Protecting Data In Use” below for additional details.

DATA SECURITY

Data security covers:

- key management,
- root of trust,
- authentication,
- access control and
- audit & monitoring.

[13] NIST Special Publication 800-38E

[14] Originally from Microsoft

[15] Transport Layer Security

Key management: Ultimately, the security of information protected by cryptography directly depends on the strength of the keys, the effectiveness of mechanisms and protocols associated with the keys, and the protection afforded to the keys. [Ref NIST SP 800-57 Pt. 1 Rev. 4] Key management refers to the operations required to manage cryptographic keys[16] throughout their lifecycle, from the time of generation to the time of destruction and at all stages in between.

The following table defines best practices associated with key lifecycle management:

Item	Key Management Best Practices
Generation	The strength of a key-generation process depends on the ability to source random numbers to produce unpredictable keys. The integrity of the random number source must be protected. For enhanced or critical security levels, only random numbers from approved cryptographic hardware should be used.
Registration	Associates the key with a user, device, system or application that will use it.
Storage	Keys must be stored in such a way that they are only accessible to authorized users or applications. Keys should be stored separately from the data they protect, and never stored in plaintext form (i.e., the keys themselves must be encrypted with different keys). Where possible and for enhanced or critical security levels, keys should be stored in dedicated, approved hardware such as Trusted Platform Modules (TPM). Where such hardware is not present, other forms of key protection such as white-box cryptography can be used to provide security.
Key creation	Devices may either: 1. Generate keys internally (using an approved source of randomness). 2. Calculate keys based on an internal physical characteristic that is in some way unique from device to device (these are called Physical Unclonable Functions (PUFs). SRAM or internal busses are examples of commercially available PUFs. 3. Accept keys from an external source in a controlled manner. Here, keys are generated externally and then sent to the device. A secure mechanism is needed to move keys from the point of generation or storage to the device or application which will use them. This requires a prior step of establishing a trusted link over which the key can be safely shared between systems. Options 1 & 2 have the advantage that the key(s) never leave the device. Option 3 has the advantage of high key quality as key provisioning systems can afford to invest in a high-quality device amortization across a high number of devices. Options 1 & 3 often require that the device allows this action only once or in very controlled circumstances.
Key use	Best efforts must be made to minimize exposure of keys in use, including controlling access to keys based on the principle of least privilege, only using

[16] Keys used for any type of cryptographic operation such as encryption and digital signing

Item	Key Management Best Practices
	keys for a single purpose, and deleting keys from working memory after cryptographic operations are completed. [Ref IEC 62443-3-3, Section 8.4.3.1] More specialized techniques for key-in-use protection, generally involve commercial tools. Examples include: • Data transformation, which changes both data and operations, allowing for computation to occur in such a way that the unprotected data never appears in memory. This makes human comprehension of the data much more difficult yet allows the application to retain its original behavior. This may be used to protect keys and also protect plaintext data after decryption. • White-box cryptography, which is a hardened implementation of a cryptographic algorithm specifically designed to either hide the key completely or keep it in a transformed state as above.
Key rotation	Keys must be periodically updated to control the risk of compromise; in consideration of the potential time it might take an attacker to guess a key based on its length. Per IEC 62443-3-3 Section 8.5.2, generally accepted practices can be found in NIST SP800-57, and implementation requirements in ISO/IEC 19790. Additionally, NIST SP800-131A Rev. 2 provides guidance on cryptographic algorithms and key lengths.
Key backup	If an encryption key is lost, the encrypted data is unrecoverable, presenting a potential availability issue. An appropriate key backup strategy ensures that only the minimum necessary number of copies of keys are created, and that their storage is properly secured.
Key recovery	When a situation arises where keys must be recovered from backups, policies and procedures must be in place to ensure that keys are restored without exposure and only by authorized parties.
Key revocation	Compromised, or suspected compromised, keys must be revoked and replaced.
Key suspension	This phase is applicable when a key is no longer being used for encryption (e.g., has expired and been rotated out), however still may be needed to decrypt operational data that has not yet been updated to the new key.
Key destruction	Keys that are no longer in use should be destroyed, including all backup copies. Destroying a key is an efficient way of destroying all the data encrypted with it, provided that all copies of the keys can be proven to be properly destroyed.

In addition to the IEC 62443-3-3 references, the content of the above table is also based on the *Key Management for Dummies* book, Thales eSecurity Special Edition (2016 update), Author John Grimm, ISBN 978-1-118-09190-6.

Cryptographic algorithms are used as building blocks to craft fit-for-purpose *modes of operation* to achieve the required cryptographic properties. For example, AES[17] is a standardized block cipher that is used as a building block to create different modes of operation. Each mode of

[17] Advanced Encryption Standard— https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.197.pdf

operation is crafted to have different properties such as secrecy or authenticity. Numerous modes of operation that been designed for use within different contexts or to have specific mathematical properties that resist various forms of attack. Sometimes algorithms are used in combination with other algorithms to create different modes of operation with multiple properties as we discuss later.

FIPS 140-2[18] defines the specific functionality required for cryptographic modules where the focus is on protecting the key material (and meta data), who can administer the keys (creation, loading, storage and destruction) and other roles around how keys are used (or consumed). The interfaces for keys, plain and cipher text are defined. Other aspects are also addressed such the design and function of the module including items such as intrusion detection and resultant actions, finite state modeling of the functionality along with other aspects such as EMC/EMI radiation and self-testing. There are four incremental security levels cryptographic modules can conform to. Level 1 focuses on software cryptographic modules and the other three demand ever increasing levels of hardware security.

FIPS 140-2 established a Cryptographic Module Validation Program[19] (CMVP), which is a joint effort between NIST and CSE that module vendors can use to validate their claims.

Each module must state the set of cryptographic algorithms, modes of operation and functionality their cryptographic module implements.

Security is difficult and users of these cryptographic modules are encouraged to ensure that only CMVP validated modules are used in their systems.

Root of Trust: Industrial IoT adds a vast number of new network-enabled devices to corporate or field networks, thereby increasing their attack surface significantly. Since IoT devices are designed for low cost with low resource consumption, it is harder to protect them using traditional methods. One of the most effective mechanisms to secure IoT devices at reasonable cost is to derive all the trust from a Hardware-based Root of Trust (HRoT).

As there are many possible attack vectors in IoT, enterprises need a solution to ensure that a particular IoT endpoint is trustworthy and safe to use for business operation. A traditional approach is to build defense in-depth with security controls at different layers of the infrastructure including endpoint, network, cloud, etc. At a minimum, IoT endpoints must have security controls to detect and possibly prevent malicious activities on the endpoints.

To enforce sound security for IoT deployments, a defense-in-depth approach should derive its trust from the HRoT on endpoints. HRoT provides secure identity, secure communications, secure

[18] Security Requirements for Cryptographic Modules FIPS 140-2 https://csrc.nist.gov/publications/detail/fips/140/2/final

[19] See https://csrc.nist.gov/projects/cryptographic-module-validation-program for more information

storage, and often a secure execution environment. HRoT also enables the device to boot securely such that only integrity-verified software can be executed on the devices.

As the device boots up, a chain of trust is established starting from boot loader, then BIOS, followed by operating system loader, operating systems, and finally applications.

There are two implementations of boot-sequence verification:

Secure boot: the HRoT verified the digest of the boot loader and compares it to the trusted digest provided by the system owner. The boot loader in turn verifies the BIOS, and so on. If digest of any component does not match the trusted signed digest, the execution sequence is stopped.

Measure boot: each component measures the digest of the component above it and stores it in a secure place for future auditing and comparison with the trusted signed digests. In measured boot, the execution sequence is not stopped even if digest of a component does not match the signed trusted value.

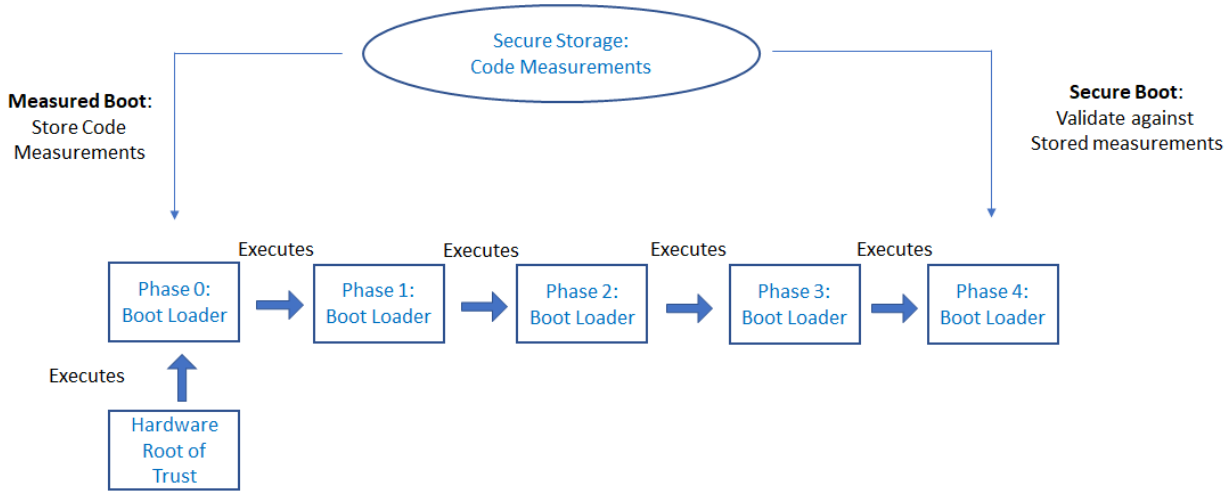

Figure 3—Secure boot and measured boot using HRoT

In secure boot, at each stage the HRoT fetches the code measures for the next stage and compares that with the instantaneous measurements. In measured boot, at each stage only instantaneous measurements are made for the next stage and they are stored in the secure storage for logging and auditing.

HRoT addresses various aspects of security in the device in different combinations. Different circuits are implemented for measurement, integrity, verification, storage and attestation purposes and used to form roots of trust that cannot be changed. This immutability is referred to as hardware roots of trust and does not reflect how it may have been implemented.

HRoT can be either implemented by a dedicated chip like a Trusted Platform Module (TPM) or processor-based HRoT. Sometimes dedicated circuitry within a processor is implemented to

protect data or software or both. Other types of HRoT protect upgradable software and implemented in external flash memory.

White-box cryptography uses mathematical techniques to hide the key in software that is normally renewable to allow developers to improve defenses over time. White-box cryptography is able to protect keys in the absence of a HRoT and can be combined with HRoTs of white-box cryptography to improve data security.

There are several commercial offerings for implementing HRoT using a processor. ARM TrustZone is a system-on-a-chip-based approach to hardware security where a secure enclave is provided for secure storage and execution. The processor is divided into normal and secure worlds. The secure world can access anything in the normal world but the normal world can only access data in the secure world through access-controlled message pipes. Memory, peripherals, execution, and interrupt are all separated at the hardware level.

Intel TEE comprises various Intel hardware-based security technologies including Intel Trust Execution Technology (TXT), Software Guard Extensions (SGX) and Platform Trust Technology (PTT). TXT improves the security of the Intel platforms from potential hypervisor attacks, BIOS or other firmware attacks and malicious rootkit installation by creating a Measured Launch Environment (MLE) that compares all critical elements of the launch environment against a trusted image.

Item	Root-of-Trust Best Practices
At rest	HRoT provides immutable secrets intended for validation of critical secure storage integrity for data at rest. This secure storage can be used to store encryption keys that are used to encrypt bulk data on the disk as well as other private or public keys that require trust from critical authentication or identity functions.
In motion	HRoT provides attestation, verification and measurement typically necessary for the establishment of secure communication channels to protect data in motion.
In use	HRoT protects data in use via authenticated secure boot processors or configuration protection of data at rest to underpin the integrity of data in use. Additional HRoT can protect individual run-time environments with secure execution environments for data within the processor and extended volatile memory if used.
Example	ARM Trust Zone can be used as a secure storage to storage security credentials and software attestation signatures in the secure partition enabled by the ARM processor. Small security applications secure boot or other critical functions that need to be tamperproof. Intel's TXT can be used to provide secure boot on the platform to ensure only trusted operating system can run on the platform.

Authenticated Encryption

Sensitive data requires encryption to prevent viewing of such data by:

- encryption's role in confidentiality and data protection,

- authentication's role is ensuring that the message is unchanged and
- recognized references for suitable algorithms and key lengths.

Encryption offers several benefits, but it can be costly. Before employing data encryption for operational data, motivations, threat vectors and general security policy should be considered.

Current cryptographic best practice is to combine authentication with encryption in an indivisible manner as this effectively defends against many attack types real world systems are exposed to. Authenticated encryption also protects the headers and other necessary information left unencrypted and is called Authenticated Encryption with Associated Data (AEAD).

Business motivations and technical motivations are often considered in isolation. The IISF provides guidance on how to consider motivations for security, including cryptographic functionality across functional roles within a company. Aligning on goals ahead of product or system design yields a more cohesive policy that will identify what should or should not be encrypted. Finally, system architects can translate the joint business and technical policies into technical requirements trading off functionality, performance, and security—including cryptographic functionality.

For example, protecting data-in-motion can be viewed at two levels: transport-level security and data-level security.

Technologies such as Transport Layer Security (TLS)[20] or Datagram Transport Layer Security (DTLS)[21] use cryptography recommended by the NSA to secure the communications transport layer, creating a secure channel between two participants. The latest version of TLS1.3 supports only AEAD ciphers. The initial set of approved ciphers is AES-GCM[22], AES-CCM[23] and the ChaCha20-Poly1305[24] combination.

Data Distributed Services (DDS) security applies the cryptography desired by system architects to data. This security is applied at the application layer so that user data can be encrypted before it is encapsulated into a transport message such that a trade-off can be made between security and performance if particular data does not require protection.

Application-level security offers the benefit of "secure multicast", which reduces the burden on the network as volumes of data increase, as illustrated below.

---

[20] Transport Layer Security (TLS) Protocol Version 1.3 https://tools.ietf.org/html/rfc8446

[21] Datagram Transport Layer Security Version 1.2 https://tools.ietf.org/html/rfc6347

[22] AES Galois Counter Mode (AES-GCM) is defined in NIST SP 800-38D

[23] AES using Counter with Cipher Block Chaining-Message Authentication Code (AES CCM) is defined in NIST SP 800-38C

[24] RFC7905 Chacha20-Poly1305 Cipher Suites for Transport Layer Security https://tools.ietf.org/html/rfc7905

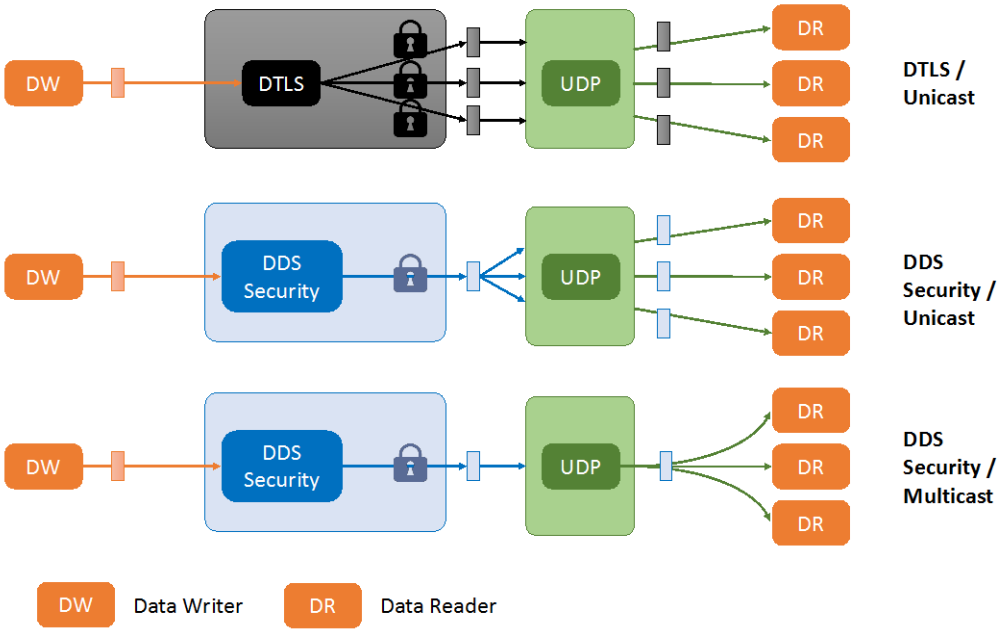

Figure 4—Data protection for Unicast and Multicast [source Real-Time Innovations, Inc. 2018]

Item	Encryption Best Practices
At rest	1. Use encrypted containers for data at rest. 2. Use NSA Suite B or other widely approved cryptography algorithms.
In motion	1. Determine data flows that need to be protected and how (digital signature, packet encryption, user payload encryption). 2. Integrate the corresponding technology that satisfies data flow, security, and performance requirements (such as TLS, DTLS or DDS Security). 3. Use NSA Suite B or other widely approved cryptography algorithms.
In use	1. See Root of Trust and Key Management. 2. Consider practice of Full Memory Encryption.
Example	Encrypting all data in a distributed control system may not be feasible due to cost, time, and performance constraints. In a healthcare setting, critical data should be considered for both encryption and authentication, but non-critical data that might be otherwise determined, such as ambient temperature may only be signed or authenticated with a Hashed Message Authentication Codes (HMAC) to optimize CPU and network bandwidth.

Session Key establishment with Mutual Authentication

The first step to protected data communications is to establish trust between the originator and recipient of data. In many cases, two endpoints may not even know the other exists. To establish trust and protected data exchanges successfully, the two endpoints must authenticate each other (“mutual authentication”) and in the process establish session key(s) that can be used to

encrypt and authenticate messages between them. When both parties are on-line the session keys should be established using cryptographic techniques that have perfect forward secrecy.

Perfect forward secrecy is the property that even if an adversary gains knowledge of the device's private key that the adversary is unable to gain access to the messages protected by session keys previously established using the now-compromised device private key. The attacker cannot impersonate the now compromised device.

To establish trust between two endpoints, use asymmetric keys with an algorithm that supports perfect forward secrecy such as Elliptic Curve Diffie Hellman (ECDH) to establish the session key. Where the session key is used in conjunction with an AEAD cipher both parties are assured that the message has not been tampered with. This significantly increases the reliability of the system.

Technologies such as TLS can be configured to use ephemeral keys[25] to ensure perfect forward secrecy. Further they could be configured to only use AEAD ciphers.

The essence of mutual authentication is the ability to prove in real time that you have access to the asymmetric private key that is paired to the public key found in the X509 Certificate. Authentication requires the use of a suitable asymmetric signature scheme. RSA (PKCS1.5 and PSS), ECDSA[26] and Edwards Signatures are all suitable pre-quantum algorithms and are currently mandated for use subject to the use of a minimum key size.

As the recommended minimum key size is under constant review, this document refrains from publishing key sizes but rather recommends that the reader either seeks professional advice or consults authoritative sources.

In July 2015 the Committee on National Security Systems released an advisory memorandum warning about the threats that advances of quantum computing pose to the breaking of asymmetric algorithms such as RSA and ECC currently in use. In July 2016 NIST[27] begun choosing new 'post-quantum' algorithms for digital signatures and session establishment. It is beyond the scope of this paper to say when quantum computing will become a risk.

[25] NIST: A cryptographic key that is generated for each execution of a key-establishment process and that meets other requirements of the key type https://csrc.nist.gov/Glossary/Term/Ephemeral-key

[26] Elliptic Curve Digital Signature Scheme

[27] NIST Post Quantum Cryptography https://csrc.nist.gov/Projects/Post-Quantum-Cryptography

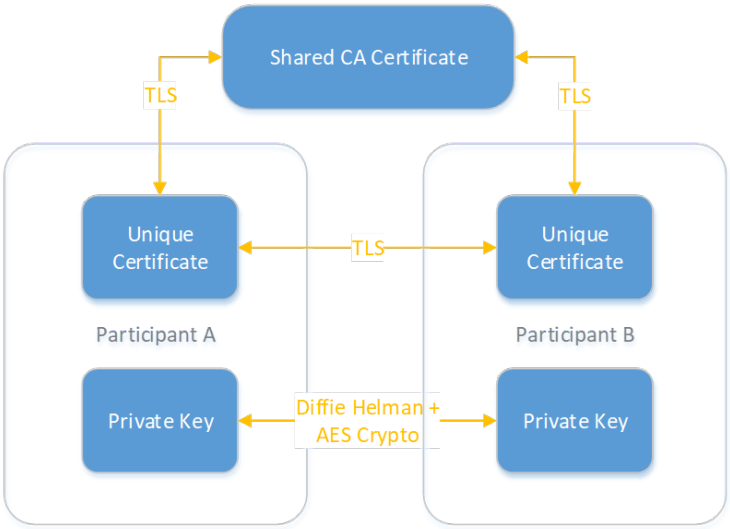

Figure 5—Example Two Factor Authentication Using ECDH [source Blackberry QNX 2018]

Item	Authentication Best Practices
At rest	See Encryption. Typically, data-at-rest is stored in encrypted containers that require authentication to decrypt the data.
In motion	1. Establish identity for network participants (users, applications, etc.) using trusted certificates and PKI. 2. Use recommended cipher suites to provide perfect forward secrecy, such as Elliptic Curve Diffie Hellman [Ref NIST SP 800-52 Rev. 2]. 3. Revoke and renew certificates according to a security policy so that trust between network participants is continually reviewed and established.
In use	For execution stack see IIC Endpoint Protection Best Practices paper.[28] For operational data: 1. Identify sensitive data to be protected. 2. Key material should be stored in hardware such as a TPM. 3. Configure a secure enclave in which sensitive data is encrypted but can still be processed by the CPU or stored in CPU cache as clear text.
Example	For in use see Root of Trust For data in motion, consider devices in a hospital clinical environment. Not all devices have a need to communicate, but those that do such as blood pressure monitor, EKG, pulse oximeter for intelligent healthcare must be able to trust each other. Devices such as pulse oximeters are available on popular online shopping sites. It would be dangerous to allow such a device to connect to the clinical network arbitrarily, but if it did, it must authenticate with the other devices before they can trust data from or send data to this new device.

[28] https://www.iiconsortium.org/pdf/Endpoint_Security_Best_Practices_Final_Mar_2018.pdf

Non-Repudiation of Information

AEAD ciphers address message security for data in motion, because both the transmitter and receiver know the mutually agreed session key. This means that either party could (in principle) alter the data and re-authenticate it, and then claim the other party made the alteration. Without additional logging or tracking data neither party can show that the other is responsible, and thus avoid liability.

There are several approaches to solving this problem including the common technique of digital signatures, where the transmitter digitally signs the data prior to transmission. To avoid the scenario of repudiating one's own signature, a Trusted Third Party (TTP) such as a Certificate Authority (CA) may be involved to objectively validate the identity of the data producer. In some situations, it is prudent to have the receiver of the data sign acceptance or send the digitally signed information to a neutral third party 'time stamping' provider. In both cases only the digital signatures must be verified to save bandwidth.

Access Control

The first step to data protection is to prohibit unauthorized access. Unauthorized access is prevented by implementing a secure authorization system and using it to enforce access control. A secure authorization system should be guided by security policies composed from organizational policies, domain security requirements, legal requirements and others.

These policies are typically written in a machine language that can be interpreted by humans. They enforce access control on all the data repositories for data at rest, on all communications channels for data in transit and for all processing applications for data in use.

The access control system serves as a reference monitor[29] for all access to data. This means that all the access paths go through the reference monitor and there are no alternate paths to access the data. For example, for data in motion, DDS accomplishes this through user-configured distributed connectivity libraries through which data passes before an application can make use of the data. TLS and DTLS accomplish this through security brokers that dictate which clients can access a particular communication channel.

Before the data can be protected, it should be categorized based on the sensitivity, such as public, restricted, confidential, secret, and top secret. The data would be labelled with the sensitivity as defined by the policy. Once the data is codified, corresponding security mechanisms are then applied to satisfy the policy.

The security policy defines the roles that can access each category of data. It also specifies the security controls to be put in place to protect each category of data from unauthorized access.

[29] A proxy that enforces security policy and cannot be bypassed.

The policy-based approach provides agility and flexibility to access control. All the components of the authorization system should be secure and should be fault tolerant with redundancy to ensure high availability.

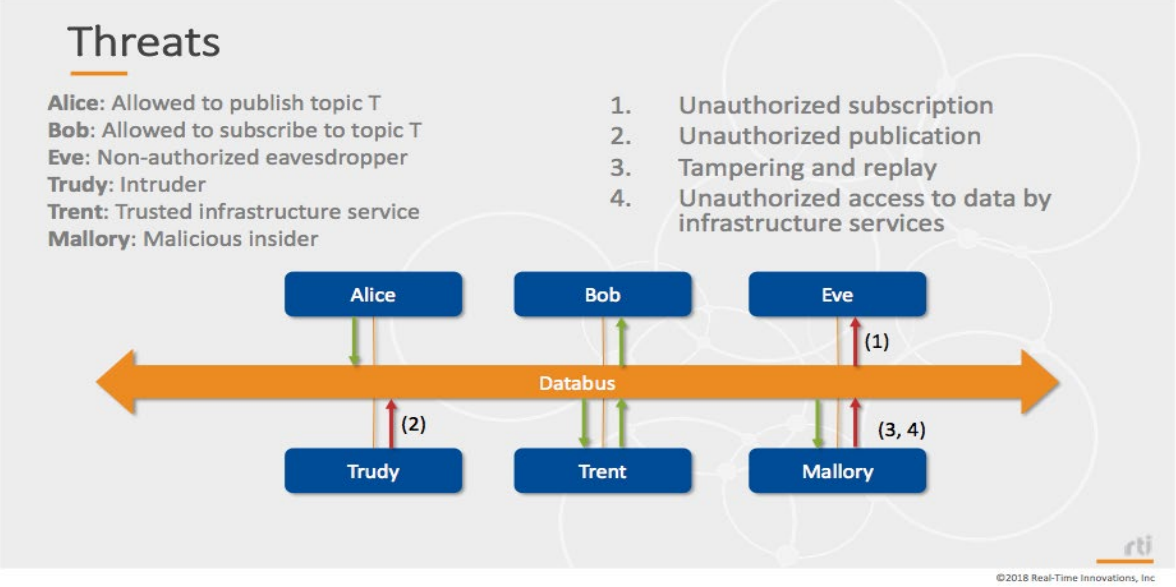

Figure 6—Threats addressed by DDS Security
[source Real-Time Innovations, Inc 2018 based on OMG DDS Security Specification]

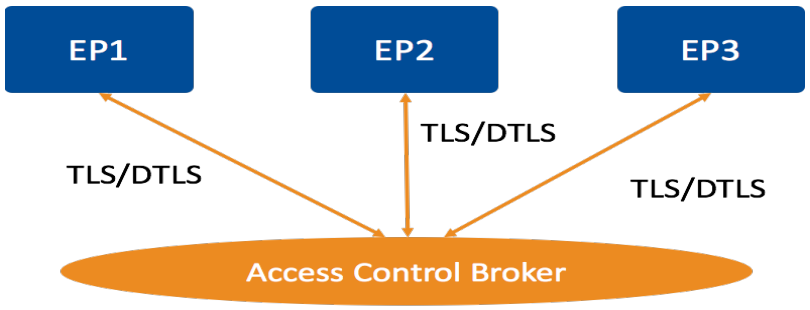

Figure 7—Access control with TLS/DTLS [Real-Time Innovations, Inc. 2018]

Note: access control for data in general works for participants operating within a defined paradigm such as DDS or a file system. Unauthorized physical entities on a network can still sniff serialized data and unpack sensitive data or unauthorized scanning of memory or hard drive may expose sensitive information.

Item	Access Control Best Practices
At rest	1. Define access control policy (whitelist and blacklist). 2. Read/write access can be defined per data entity (i.e. file) or per container (i.e. database or storage container). 3. An automated mechanism should enforce access policies.
In motion	Similar to at rest.

Item	Access Control Best Practices
In use	Similar to at rest. Use a separation kernel for time and space partitioning to ensure sufficient isolation of data and processes.
Example	Not all devices on a network are entitled to the same behavior with regards to data. For example, in a healthcare setting, an infusion pump application should have read access to medication commands but be denied write access to patient data such as heart rate or temperature, which are captured by other devices.

Audit and Monitoring

The system must be monitored to understand the current security state of the system, validate that the system is operating as intended, that no policy violations have occurred, and that there have been no incidents. In the absence of system monitoring, system operators will not be able to determine the security state, nor whether these has been an attack on the system.

Monitoring can be implemented either using an agent that monitors system events and reports Indicators Of Compromise (IOC) to the security operations center, or by exporting systems logs to a log aggregation system where they can be analyzed for IOCs. System monitoring, logging and auditing is needed to troubleshoot and perform forensic analysis of the system.

Monitoring data access is a challenge in low-bandwidth and high-cost communications channels such as oil and gas located in remote areas. Transmission of security logs has an associated cost and it competes with transmission of operations data on the same low-bandwidth link so security logs should be pre-processed and compressed to ensure optimal use of available resources.

Information related to the security event must not be lost in this preprocessing because reconstruction of the event would become impossible. This can be addressed by storing system logs locally and making offline backups during regular maintenance visits.

Item	Audit and Monitoring Best Practices
At rest	Data at rest can be monitored for access policy violations and logs should be created for security audits. IOCs should also be monitored for unauthorized access detection.
In motion	Data in motion should be monitored for ingress and egress policies. Logs should be created for security audits. IOCs should be monitored for unauthorized access detection.
In use	Data in use should be monitored for unauthorized access or privacy requirements.
Example	An example of monitoring industrial IoT devices is monitoring ICS and SCADA devices in the Operations Technology (OT) environment. These devices control complex, high-value and mission-critical systems. To ensure that these devices are not compromised, and the networks are not breached, endpoints and networks should be monitoring continuously to detect any anomalous security event in the system. These security events should be analyzed either onsite or in the cloud in the context of the domain and any potential security

Item	Audit and Monitoring Best Practices
	incidents should be reported in the form of security alerts to security operations center. These security events should be archived for auditing and forensic analysis of security incidents during an investigation.

Protecting Data-in-Use

There is no standardized approach to keep data protected while simultaneously attempting to use it. Access Control (see above) seeks to limit who can access data. Other techniques focus on limiting where the data can be accessed, specifically trying to ensure that it does not appear in unprotected computer memory.

Trusted Execution Environments (or TEE, see HRoT above) perform computations in a separate area, with dedicated and protected memory and processing. Where available, an HRoT effectively limits access because only the TEE can see the data in use.

Data transformation involves software-based techniques that modify the data to be used and the corresponding operations performed on that data. The data remains transformed throughout its use. In other words, the unprotected value never appears, and all computational operations occur on the data in protected form. This is in contrast to masking methods that are temporally removed to reveal the unprotected data prior to the computational operation. Data transformation makes it harder for adversaries to perform run-time analysis of software and data as the actual functionality is hidden.

White-box cryptography is a specialized form of data transformation applied to the data (keys and plaintext/ciphertext) and operations of a cryptographic algorithm. When the encryption/decryption key is known in advance, the operations are changed to pre-incorporate the key, thus removing it as a dynamic input and completely removing it from memory. In another form, the key is loaded in a transformed state and the operations are modified to work on the transformed key. A developer might put a placeholder for the white-box cryptographic implementation in their code, which would be instantiated with the real implementation when building the application.

In many cases, a combination of techniques may be needed, along with strong access control, and auditing to have forensic information if and when breaches occur.

DATA INTEGRITY

Data integrity refers to maintaining the accuracy and validity of data throughout its lifecycle, ensuring that it is not altered or destroyed in an unauthorized manner. In industrial

environments, data integrity and system integrity are closely related, as manipulation of industrial systems and communication channels can directly result in a loss of data integrity.[30]

The following aspects of data integrity must be considered:

Accuracy of data: operational or business decisions based on inaccurate data will likely be faulty.

Timeliness of data: industrial systems in particular are sensitive to latency; full-time network connectivity may not be guaranteed, and data may be incomplete or missing.

Protection of data against tampering: data manipulation by human operators or by systems corrupted with unauthorized software can disrupt operations and result in safety concerns.

Introduction of data errors: unintentional errors can originate from several sources, such as human operators, faulty communication protocols, and misconfigurations.

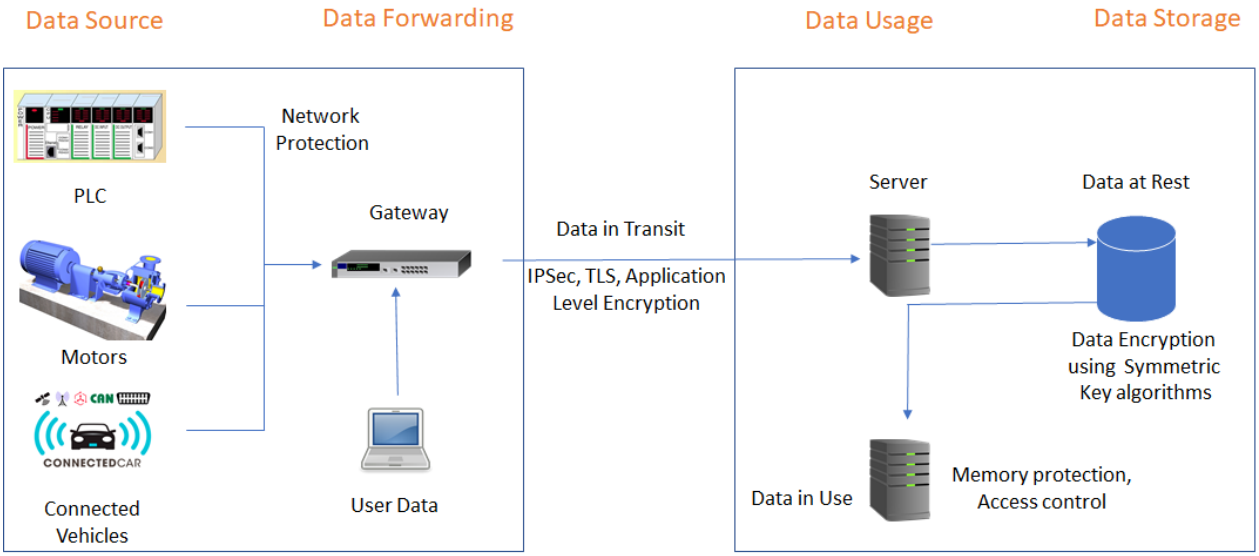

Figure 8—Schematic diagram of data lifecycle and protection mechanisms

Data integrity can be violated intentionally by malicious actors or unintentionally by corruption during communication or storage. Data integrity assurance is enforced via cryptographic controls for detection of integrity violations. The actual control depends on the lifecycle phase of the data.

Data integrity can be classified as physical integrity or logical integrity depending on whether the cause of integrity violation is physical issues with storage or transmission or logical issues with data analysis.

[30] References: IEC 62443-3-3, NISTIR 8222 (Draft)

The figure above shows a data lifecycle where the data passes through multiple phases from generation to long-term storage. When the data is produced in an industrial environment, it is usually sent to a gateway in the same network over a variety of ICS, SCADA or internet protocols. Data integrity is at risk when insecure protocols are used; network protection is recommended. Network isolation and network access control are additional security controls that can improve data integrity in this environment.

IoT installations generally have an edge device that enables communications with external networks using standard protocols. Various radio-access technologies are used to connect to external networks. Data in transit over standard internet protocols should use standard network security including IPSec or TLS.

For highly sensitive data application-level cryptographic processing can provide confidentiality and integrity for data in motion as discussed in the section on AEAD ciphers. When data is stored in long-term data storage, it should be encrypted for confidentiality and integrity protection.

When data is in use, it must not be modified in an unauthorized manner. Access to memory locations and registers must be controlled.

When data is sent from a producer to a consumer, its integrity should be verifiable by creating a secure digest of data at the origin that can be verified at the destination, using cryptographic algorithms such as digital signatures using asymmetric keys or HMAC using symmetric keys. The secure digest accompanies the data during its lifetime. Before the data is used by the consumer, its secure digest should be verified to ensure that data integrity is intact.

Data Security Perspective

We distinguish between data confidentiality and data integrity:

Data confidentiality concerns the protection of sensitive, secret, or proprietary information from disclosure, and is typically provided by cryptographic mechanisms such as encryption. Encryption limits the ability to read that information to those in possession of the encryption key, thus protecting the secrecy of that data.

Data integrity means immutability, or protecting against alteration, rather than secrecy. A reading from a temperature sensor on industrial equipment may not be considered secret and therefore not worthy of encryption in a particular environment, but if altered to indicate an outside-normal value, might trigger an alarm and disrupt operations. Cryptographic means, such as message authentication codes and digital signatures, are useful to ensure integrity of data.

To protect the integrity of data, the following requirements apply:

Item	Data Integrity Best Practices
At rest	The integrity of data at rest must be ensured by standards-based mechanisms, such as backup and replication. Mechanisms must be provided to monitor configuration data of systems to detect unauthorized changes or directly to protect critical data from non-authorized alterations (malicious or accidental).
In motion	The integrity of data in motion must be ensured by employing protocols that use standards-based cryptographic algorithms and methods to ensure modifications or alterations are detected.
In use	The integrity of data-in-use must be ensured by mechanisms such as policy-based authorization and access control for users and applications, and trusted and secured memory for execution. Mechanisms must be provided to prevent and detect any attempted introduction of unauthorized software (e.g., viruses, malware) to systems that produce and operate on IoT data.
Example	The integrity of data-in-motion is ensured by using standard secure communications using TLS 1.2 or higher to protect the confidentiality and integrity at the network level. Message Authentication Codes (MACs) can be implemented at the application protocol layer to provide message integrity checks. Two types of MAC are in widespread use. HMAC makes use of digital hash functions like SHA2 or SHA3 families while CMAC uses a block cipher like AES.

Data integrity is a common topic across many industries. As such, the best practices are usually developed by standards organizations and then adopted for industries. NIST is creating data integrity standards via its National Cybersecurity Center of Excellence.

The project is creating three guidance documents:

- Data Integrity: Identifying and Protecting
- Data Integrity: Detecting and Responding
- Data Integrity: Recovering

These documents are aligned with NIST Cybersecurity practice guide SP 1800-11,[31] Data Integrity: Recovering from Ransomware and Other Destructive Events.

Best practices for responding to integrity breaches and recovering from integrity breaches are beyond the scope of this white paper. Interested readers should refer to the NIST guidance document for details on these topics.

[31] https://csrc.nist.gov/publications/detail/sp/1800-11/draft

SECURING THE IOT INFRASTRUCTURE

Securing an IoT infrastructure requires a rigorous security-in-depth strategy that:

- secures data in the cloud,
- protects data integrity while in transit over the public internet and
- securely provisions devices.

The security-in-depth strategy should be developed and executed with active participation of various players involved with the manufacturing, development and deployment of IoT devices and infrastructure.

The table below lists the players involved in developing a security-in-depth strategy for an IoT infrastructure and summarizes that strategy.[32]

Players involved	Security-in-depth Strategy	Notes
IoT Hardware Manufacturer/Integrator	• Scope hardware to minimum requirements • Make hardware tamper proof • Build around secure hardware • Make upgrades secure	Address security requirements to address storage and flow through.
IoT Solution Developer	• Follow secure software development methodology • Choose open-source software with care • Integrate with care	
IoT Solution Deployer	• Deploy hardware securely • Keep authentication keys safe	
IoT Solution Operator	• Keep system up-to-date • Protect against malicious activity • Audit frequently • Physically protect IoT infrastructure • Protect cloud credentials	

Capabilities of different IoT devices vary, for example computers running common desktop operating systems and devices that run light-weight operating systems. If provided, additional security and deployment best practices from the manufacturers of these devices should be followed.

Some legacy and constrained devices might not have been designed specifically for IoT deployment, and thus may lack the capability to encrypt data, connect with the internet, or provide advanced auditing. In these cases, modern and secure field gateways that can provide

[32] Microsoft Security best practices for Internet of Things https://docs.microsoft.com/en-us/azure/iot-fundamentals/iot-security-best-practices

secure authentication, negotiation of encrypted sessions, etc. can aggregate data from legacy devices and provide the security required for connecting these devices over the internet.

DATA PROTECTION AND IOT TRUSTWORTHINESS

The IIC IoT Trustworthiness framework embodies the viewpoint that IoT is more than just "IT for Things". For an IoT system to operate in conformance with business and legal requirements, several characteristics of that system, namely security, safety, reliability, resilience and privacy, must remain compliant with these requirements, despite environmental disturbances, human errors, system faults, and attacks.[33]

Data security (and thus data protection) plays a central role in the enablement of IoT trustworthiness and its characteristics: privacy, reliability, resilience, and safety. We explore now the role of data security in a system safety context.

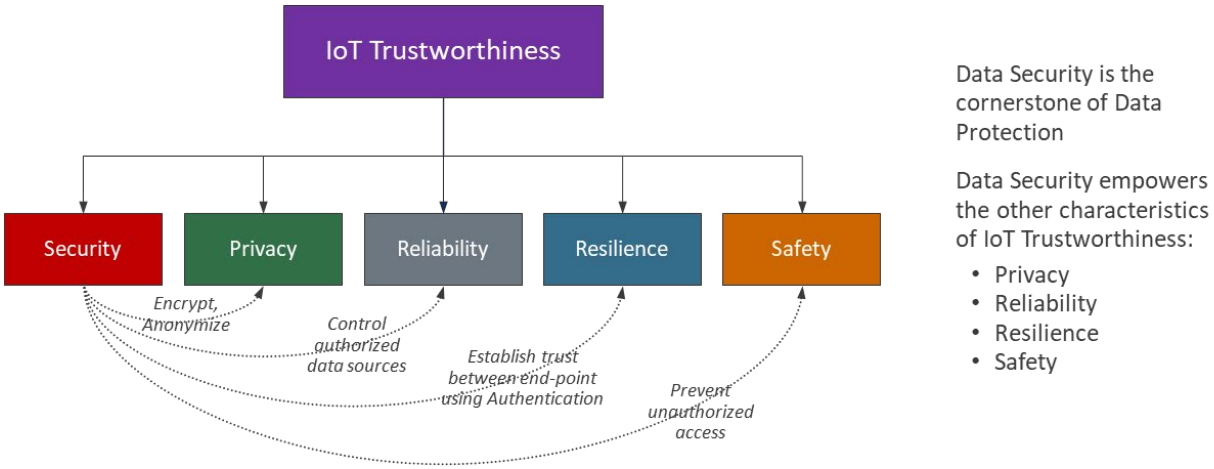

Figure 9—Data protection empowers IoT Trustworthiness [source IGnPower]

Role of Data security in a System Safety context

Industrial safety[34] systems have well-defined standards that govern them. Data protection for these systems also come in their purview, since manipulation of this data could lead to safety

[33] The IIC Journal of Innovation, 9th Edition: Trustworthiness. 2018-September. www.iiconsortium.org/news/journal-of-innovation-2018-sept.

[34] Key Safety Challenges for the IIoT, IIC Technical White Paper IIC:WHT:IN6:V1.0:PB:20171201, 2017-December-01, https://www.iiconsortium.org/pdf/Key_Safety_Challenges_for_the_IIoT.pdf

incidents. IEC 61508[35] [36] is one such standard that applies to functional safety of Electrical, Electronic and Programmable Electronic Safety-Related Control Systems [37] (EEPE), which constitute the electronic controllers for industrial IoT.

The standard covers the entire safety lifecycle and uses a probabilistic risk-based approach to system safety, that accepts the view that risks can only be minimized, but not eliminated. Minimizing non-tolerable risks results in an optimal, cost-effective safety program.

The standard targets a safety integrity level (SIL) for each safety function based on a risk assessment that considers the risk to safety-critical data generated and consumed by the system. Data protection techniques to protect confidentiality, integrity, and availability of safety-critical data, and establishing its authenticity, while providing non-repudiation capability are important parts of this risk assessment.

The risk assessment should consider the impact of any missing data-protection feature on the system. The impact and likelihood of exploitation determine the risk and its severity. The severity would also indicate what level of security effort is justified to address that risk.

This standard also recommends the separation between control and safety systems. This places further burdens on the methods implemented for data protection, since data protection methods must now affect the safety systems and not only the control systems.

Item	Data protection Best Practices in a Safety Context
At rest	Similar to in motion (below). Protecting the integrity of the IoT data-at-rest helps ensure the system operates in the correct states safely.
In motion	Ensure mutual authentication of endpoints. Safeguard the integrity of IoT data to ensure proper safety functions of the system.
In use	Similar to in motion. If a safety-critical system processes data that is altered to be outside of safety parameters, the integrity of the system could fail.
Example	In IoT use cases where control systems and safety systems must be separated as per IEC 61508, specific data protection measures must be applied to IoT data that can affect safety. For example, system commands sent to the process control or manufacturing systems determine how the operating state of the system would change. If an attacker can change the message field in the commands, he could push the system into dangerous state.

[35] www.iec.ch/functionalsafety/

[36] https://www.iiconsortium.org/pdf/Industrial_Internet_of_Things_Volume_G2-Key_System_Concerns_2018_08_07.pdf—IIC Industrial IoT—Key System Concerns G2 section 3

[37] http://www.iloencyclopaedia.org/part-viii-12633/safety-applications/94-58-safety-applications/electrical-electronic-and-programmable-electronic-safety-related-control-systems

Item	Data protection Best Practices in a Safety Context
	Also, system data context is vital. A real-life example occurred when NASA lost its $125M Mars Climate Orbiter because the team of spacecraft engineers failed to convert data from the imperial measurement system to metric measurement systems causing the spacecraft to crash.[38]

Data protection (especially security) plays a key role in other trustworthiness characteristics, specifically system reliability and system resilience.

OTHER DATA PROTECTION CONSIDERATIONS

This section provides a few examples of domains and other topics that are associated with data protection (and requiring the use of data security mechanisms), namely:

- data privacy,
- data confidentiality,
- data residency,
- data electronic discovery (ediscovery) and legal holds,
- security-level markings and
- data lifecycle management.

Data Privacy

Personal data, for example in healthcare and smart cities, must be protected in accordance with applicable data privacy laws and regulations.

According to the International Association of Privacy Professionals,[39] "Data privacy is the appropriate use of personal information under the circumstances. What is appropriate will depend on context, law, and the individual's expectations; also, the right of an individual to control the collection, use and disclosure of information.

"This refers to the right of individuals to control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed".[40]

Data privacy regulates how personal data should be handled, what types of actions can be carried out on that data, and who is authorized to carry out these actions.

[38] https://mars.jpl.nasa.gov/msp98/orbiter/

[39] IAPP https://iapp.org/

[40] ISO 7498-2:1989 https://www.iso.org/obp/ui/#iso:std:iso:7498:-2:ed-1:v1:en

Data privacy laws are proliferating in various jurisdictions,[41] and they are becoming increasingly stringent. Thus the need for compliance with these laws is becoming a major concern.

The terms “privacy by design” and “privacy by default” refer to a systems-engineering approach that calls for privacy to be taken into account throughout the lifecycle of the system.[42]

A prime example of data privacy laws is the EU General Data protection Regulation[43] that grants data subjects a wide range of rights over their personal data, namely the right to:

- know what is being done with their data (consent),
- have their incorrect data corrected,
- have their data forgotten (data erasure),
- have restrictions on processing of their data (clear scope),
- have data portability (withdrawal of data) and
- object to their data being processed (adapting to changing situations).

These rights translate into explicit restrictions on organizations (data controllers and data processors) that handle personal data within the EU jurisdictions and personal data belonging to EU residents outside EU jurisdictions.

The figure below provides an outline of these restrictions:

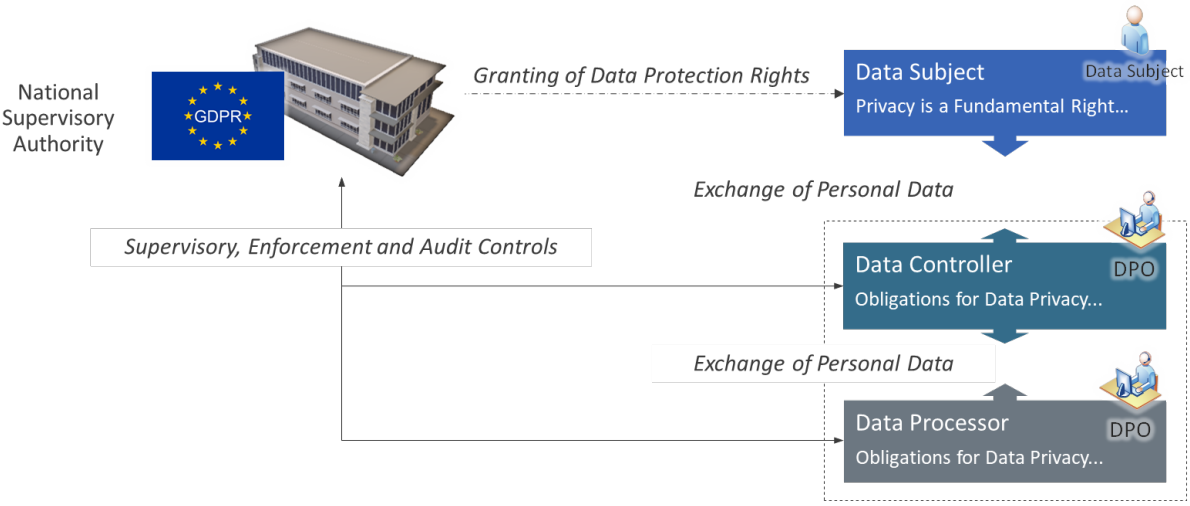

Figure 10—GDPR Data protection Process [source IGnPower]

41 GDPR in the EU, CCPA in California, PIPEDA in Canada.

42 Refer to the definition of PbD in the Acronyms section.

43 GDPR https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679&qid=1473816357502&from=en

For IoT solutions to comply with the GDPR requirements, they must leverage data security mechanisms, for example key management, authentication and access control. Refer to the Data Security section in this document.

DATA MINIMIZATION

Personal data collected from data subjects must be reduced to the lowest levels necessary for the specific purpose of the processing. The scope and amount of data collected is use-case specific, and must be adequate, relevant and limited to the scope of the transaction.

Item	Data Minimization Best Practices
At rest	1. Identify scope of personal data to be collected. 2. Validate that scope against the requirements of privacy laws. 3. Implement data collection methods which, by design, should minimize the collection of personal data. 4. Minimize the collection of personal data, based on requirements of use case. 5. Encrypt personal data while in storage (after processing has been completed).
In motion	Encrypt personal data while in transit from collection point to processing point.
In use	Impose restrictions on usage of personal data within the IoT solution and beyond.
Example	GDPR recognizes that personal data may need to be collected from a data subject to satisfy the scope of a particular use case. The scope of these use cases and the personal data must be analyzed, validated and documented. The collection of the personal data must incorporate the data-subject consent process. In case of data breaches, specific breach protocols (including notification) must be enacted within specific timeframes.

DATA ANONYMIZATION

Data anonymization is typically used for long term dissemination of information where the focus is on the information and not the data subject, for example in medical research. It is a data-masking method that permanently replaces data deemed to be personal with generalized or randomized data so that what remains has no connection back to the data subject. Data anonymization can take place during storage and during output. Specifically:

- identify scope of personal data to be anonymized,
- define anonymization strategy: replace with generalized or randomized code, for example <masked_data> or replace specific code based on data type, for example <patient_ID>,
- implement the anonymization process on said data and
- ensure that the anonymization action has been implemented at all the layers of the architecture stack, down to the physical storage layer (if required)

Item	Data Anonymization Best Practices
At rest	By definition, anonymized data is not subject to data privacy controls—it is up to the organization's legal department or data privacy officer (DPO) to make that determination.
In motion	Similar to at rest.
In use	Similar to at rest.
Example	It is the organization's data protection officer who decides whether personal data that has been properly anonymized can be exempt from the GDPR controls.

DATA PSEUDONYMIZATION

Data pseudonymization is a data-masking method that replaces data deemed to be personal with encoded or masked data that cannot be traced back to the data subject without using additional information. The relationship between the encoded or masked data and the original data subject must be stored separately in highly secure environments.

Item	Data Pseudonymization Best Practices
At rest	1. Identify scope of personal data to be pseudonymized. 2. Define pseudonymization strategy: replace with generic code or with specific code by data type. 3. Implement pseudonymization processes on said data. 4. Ensure that anonymization action is implemented at all layers of the architecture stack, down to the physical storage layer (if required). 5. Store mapping between original personal data and pseudonymization code in a secure environment (same restrictions as restrictions in personal data).
In motion	Apply similar security controls to confidential data.
In use	Apply similar security controls to confidential data.
Example	Under GDPR, pseudonymized data are subject to privacy controls and restrictions since the original personal data can be reconstructed if the mapping between this personal data and the pseudonymization code is accessed.

Pseudonymized data can in principle be exchanged with other parties without a privacy breach, but because the relationship between the encoded or masked data and the original data subject can be re-established, pseudonymized data must be subject to privacy controls.

Data Confidentiality

A data breach may affect data that is business-sensitive in nature, and the loss of this data may lead to one or more types of losses, for example:

- business: revenue loss, loss of competitive advantage,
- financial: profit loss, non-compliance fines,

- litigation: financial exposure and criminal exposure[44] and
- reputational damage.

Item	Data Confidentiality Best Practices
At rest	1. Identify scope of data deemed to be confidential in nature. 2. Determine level of restrictions on access of such data. 3. Implement methods to label data as being confidential. 4. Implement security methods and mechanisms as described in the data security section, for example encryption, authentication and access control.
In motion	Encrypt confidential data while in motion from collection point to processing point.
In use	Implement security methods and mechanisms as described in the data security section, for example encryption, authentication and access control.
Example	Similar mechanisms and techniques to data privacy can be used with data confidentiality, for example data encryption, data minimization, data anonymization, and data pseudonymization. Moreover, data confidentiality settings may need to change during the lifecycle, for example the configuration and performance data of a system or component may be embargoed prior to its release to the market.

In most government organizations,[45] data and information are assigned security levels that correspond to their sensitivity vis-à-vis the potential damage to national security if released, for example Unclassified, Confidential, Secret, and Top Secret, with additional restrictive filters applying at the high-end of the scale such as NOFORN.[46]

Depending on the use case involved, the principles of security-level classifications and the methods and mechanisms to implement them can also apply to IoT data. This means that a security strategy[47] must be developed and executed to assign appropriate data access rights to sensitive data based on the security classification markings assigned to the data versus those assigned to the users. Users are authorized to access data and information that have security classification labels up to their own level and not beyond.

Security-level classifications may also apply to the systems themselves where the data is stored or flows through. For example, a system accredited for Secret data may not be authorized to receive or host Top Secret data or let such data flow through it. This is because a system accredited for Secret data will most likely lack the data protection capabilities that are necessary for the protection of Top Secret data.

---

[44] Due to spoliation, which is the “destruction” of evidence.

[45] Some corporations, for example major financial institutions, have adopted a similar security classification schema.

[46] No Foreigners

[47] Access rights, encryption, etc.

Security declassification processes are beyond the scope of this paper.

Data Residency

Laws and regulations may mandate that certain types of data remain within the boundaries of specific jurisdictions. Data residency restrictions are common with financial data and health data.

Data residency[48] restrictions can also apply to critical data (industrial secrets, IP, etc.) and must remain within the jurisdiction at all stages of its lifecycle.

Item	Data Residency Best Practices
At rest	1. Certain laws and regulations may not allow the storage of data of specific data types outside the jurisdiction. 2. Create classification of data types. 3. Identify data types that have data residency requirements. 4. Classify data by source and data type. 5. Design IoT system whereby data that has data residency requirements is stored in locations that are physically situated within the jurisdiction. 6. The legal and compliance departments may opine that certain data types are subject to data residency restrictions, though they may be stored outside the jurisdiction if they are encrypted.
In motion	The above restrictions apply to data in-motion in the same manner. Other restrictions such as data privacy and data confidentiality may still apply.
In use	The above restrictions apply to data-in-use in the same manner. Even if data with data residency restrictions is encrypted and stored outside the jurisdiction, it cannot be used outside the jurisdiction. Other restrictions such as data privacy and data confidentiality may still apply.
Example	Most laws regulating the use of electronic personal health information, for example US HIPAA[49] & HITECH, UK NHS, mandate that personal health information must be located within the national jurisdictional boundaries .

Data eDiscovery and Legal Holds

The process of legal ediscovery requires the use of data protection methods. Industrial accidents and other unwelcome events can lead to investigations and possibly legal action. Configuration data and data produced and consumed by these systems can be relevant to pursuant

[48] Issues and practices related to the location of data and metadata, the movement of (meta)data across geographies and jurisdictions, and the protection of that (meta)data against unintended access and other location-related risks.

[49] US Health Insurance Portability and Accountability Act https://www.hhs.gov/hipaa/index.html

investigations or litigation, so they will be deemed Electronically Stored Information (ESI), subject to eDiscovery and legal-hold processes.[50]

Item	eDiscovery and Legal Holds Best Practices
At rest	1. Define scope of data that is found to be "responsive" to the scope of the investigation or litigation: data range, data values, phrases, etc. 2. Search for and locate IoT data that is "responsive"[51] to scope. 3. Collect IoT data[52] and metadata, after validation of authenticity. 4. Deliver collected data to the legal and investigative teams. 5. Place the IoT data on hold (revoke all rights to edit or delete this data). 6. Where required implement recurring integrity checking methods to establish immutability (legally defensible). 7. Release the hold on the IoT data when this data is deemed to be no longer needed for investigation or litigation. 8. Maintain audit trail about above.
In motion	Above data may be personal or confidential in nature. The data-protection methods implemented for personal data and confidential data should apply to this data while in-motion.
In use	Same restrictions as with in motion.
Example	EDRM.net[53] is a widely accepted electronic discovery framework. It provides a conceptual view of the eDiscovery process and the different steps involved in it. Some of these steps require the use or core data security functions, for example immutability and integrity.

Data Lifecycle Management

Data must have a finite life. It is created, stored, consumed, exchanged and eventually disposed of once the operational, archival, legal and regulatory retention requirements have been complied with.

Data volumes are exploding, and the decision about how long to retain this data and when to delete it must consider:

- IoT systems have long lifecycles: design, build, operate, maintain, decommission,

[50] Collection and preservation (including context) in a manner that guarantees authenticity, integrity and chain of custody (lineage).

[51] Federal Courts Law Review - Glossary of Technology-assisted Review: Document that is Relevant to an Information Need expressed by a particular request for production or subpoena in a civil, criminal, or regulatory matter. https://www.fclr.org/fclr/articles/html/2010/grossman.pdf page 28

[52] The source of the data may include the IoT system tiers and the audit trail

[53] Electronic Discovery Reference Model http://www.edrm.net/: Collect and preserve data (including context) in a manner that guarantees authenticity, integrity and chain of custody (lineage).

- operational needs and value of IoT data,
- analytics value of IoT data: preventive, predictive, prescriptive and
- legal and regulatory obligations towards IoT data.

IoT data can become a liability if it is over-retained beyond what is "necessary", not least in terms of lower performance and increased infrastructure costs.[54] Over-retaining data can also heighten legal and regulatory risks due to the fact that the organization must now protect higher volumes of data than what is necessary or required against leaks, privacy breaches, litigation etc.

On the other hand, disposing of IoT data prematurely (especially historical data) can lead to the loss of valuable assets and insight.[55] It can also increase regulatory non-compliance risks[56] as well as legal risks and costs.[57]

Data-lifecycle management is a balancing act that must be performed systematically in accordance with established up-to-date policies, and with legally defensible audit trails.

Best practices for data lifecycle management must cater for changes in policies and how they impact data during the lifecycle, for example changes in data confidentiality requirements or changes in data residency requirements.

Data lifecycle management relies heavily on data protection mechanisms, especially data security. For data to be trustworthy during its lifecycle, it must be retained in a manner that is immutable to ensure its security, integrity, context, and lineage (provenance and chain of custody). When the data expires, the end-of-life (EoL) disposition action(s) must be triggered. This is the action prescribed for that type[58] of data in the lifecycle policy.

[54] Despite the myth, storage is not cheap once the overall infrastructure, communication, and organizational overheads are taken into consideration.

[55] Rapidly advancing AI analytics tools require big data to produce valuable insight.

[56] Non-compliance with mandatory data-retention requirements.

[57] Spoliation (destroying evidence) if the data subject to litigation is deleted.

[58] Example: temperature readings, pressure readings, location.

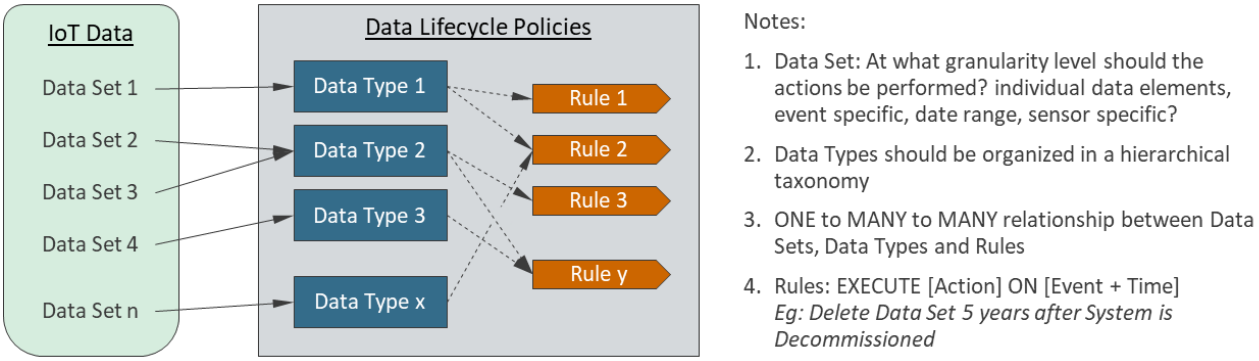

Figure 11—Data Lifecycle Policies [source IGnPower]

The action on the data defined in the EoL point of the policy may be any of the following:

- delete data,
- transfer legal custody of data to another authority or entity,[59]
- declassify data and then transfer it and
- sanitize data (anonymize, pseudonymize) and then transfer it.

Data deletion may be triggered based on pre-defined durations that may start when the data is created or captured or when an external event occurs.

Some use cases may require data deletion to be enforced at all layers of the storage stack, [60] including the physical storage medium. This may require the use of special system tools that can erase or overwrite data from the physical storage medium itself.

Item	Data Lifecycle Management Best Practices
At rest	1. Define classification of unique data types produced and consumed by the system. 2. Identify legal and regulatory lifecycle and disposition requirements for data type. 3. Define lifecycle policies for data types (including EoL actions). 4. Implement methods to enforce integrity and immutability on data during the lifecycle of the data (based on policies associated with each data type). 5. Implement methods to trigger and enforce EoL actions on data at the end of their lifecycle (based on established policies). 6. Generate legally defensible audit trail about above.
In motion	The design of the IoT system architecture stack must identify the segments of that architecture when and where IoT data can exist in motion. It must also ensure that this data is not retained permanently in these segments.

[59] Example: US Federal Agencies “accessioning” of data to NARA (US National Archive)

[60] US DoD refers to such actions as “expungement”.

Item	Data Lifecycle Management Best Practices
In use	Similar to the in-motion requirement, the design of the system architecture stack must identify the segments where IoT data can be in use. It must also ensure that this data is not retained in these segments.
Example	The EU GDPR right-to-forget mandates that certain personal data be deleted when the data subject requests such an action.

CONCLUSION

Protecting IIoT data is one of the critical foundations of trustworthy systems. Data protection measures resist internal and external disturbances and attacks on critical data and IIoT systems at large. Failure to apply such measures to data at rest, data in motion, and data in use can lead to serious consequences for IIoT systems, such as service disruptions, serious industrial accidents, loss of IP, regulatory fines, and negative impact on brand reputation.

In this paper we have identified security of data as the cornerstone of Data Protection. So we have focused on the measures and best practices needed to achieve a desired level of security for data, for example key management, root of trust, authentication, access control and audit & monitoring. We have also described the best practices for several domains that fall under the umbrella term of data protection, for example Data Privacy, Data Confidentiality and Data Integrity, as well as a number of adjacent domains that rely on security mechanisms to achieve their objectives, for example Data Residency and Electronic Discovery and Holds.

The viewpoint that IoT is much more than just “IT for Things” is embodied within the IIC IoT Trustworthiness framework. The paper also covers the best practices for data security within the context of trustworthiness and its individual OT-related characteristics, especially safety.

The Data Protection Best Practices paper complements the IoT Security Maturity Model [IISM) and builds on the concepts of the Industrial Internet Reference Architecture [IIRA] and Industrial Internet Security Framework [IISF].

GLOSSARY

Term	Explanation
AEAD	Authenticated Encryption with Associated Data. Parts of a communication message is left in the clear (i.e. un-encrypted) to allow for message routing, message typing and for the transfer of other meta data required to process or store the message with deciphering the message. This clear data is called ‘Associated Data’ in the cryptographic community.
HRoT	Hardware Root of Trust. Where the security functionally is implemented in an unchangeable manner. As the name implies this is typically implemented using logic gates but can also be implemented in unchangeable software (often referred to as firmware).

Term	Explanation
SRoT	Software Root of Trust. Where the security functionality is implemented in a changeable or upgradable manner. As the name implies this is usually done in software but similar functionality could be implemented using FPGA logic.
PbD	Privacy-by-design is an approach to systems engineering initially developed by Dr. Ann Cavoukian and formalized in a joint report on privacy-enhancing technologies by a joint team of the Information and Privacy Commissioner of Ontario (Canada), the Dutch Data Protection Authority and the Netherlands Organisation for Applied Scientific Research in 1995. The framework calls for privacy to be taken into account throughout the whole engineering process. The concept is an example of value-sensitive design, i.e., to take human values into account in a well-defined manner throughout the whole process.
SbD	Security-by-design is an approach to building systems that incorporates security in each function of the system right from the ideation to completion of the product. The goal is to eliminate vulnerabilities from the systems and make it resilient to attacks. Security-by-design relies on a comprehensive, secure development lifecycle (SDLC) to include security requirements that provide guidance for developing secure architecture and design. The SDLC also ensures that the design is developed into product security by using secure development guidelines for hardware and software. For software systems, secure coding practices along with static and dynamic code analysis is performed. Once the system is developed, security testing should be performed to ensure that the system complies with the defined security requirements.

REFERENCES

[IIC-IIRA2015] Industrial Internet Consortium: The Industrial Internet, Volume G1: Reference Architecture Technical Report, version 1.7, 2015-June-04
http://www.iiconsortium.org/IIRA.htm

[IIC-IIV2015] Industrial Internet Consortium: The Industrial Internet, Volume G8: Vocabulary Technical Report, version 1.0, 2015-May-07
http://www.iiconsortium.org/vocab/index.htm

[IIC-IIV2018] Industrial Internet Consortium: The Industrial Internet of Things—Volume G8: Vocabulary Version 2.1, IIC:PUB:G8:V2.00:PB:20180822, 2018-August-22
http://www.iiconsortium.org/vocab/index.htm

[IIC-IISF2016] Industrial Internet Consortium: The Industrial Internet of Things Volume G4: Security Framework Version 1.0, 2016-September-26
http://www.iiconsortium.org/IISF.htm

[IIC-SMMD2019] IoT Security Maturity Model: Description and Intended Use, IIC:PUB:IN15:V1.0:PB:20180409, 2018-April-09

	http://www.iiconsortium.org/IISF.htm
[IIC-SAFETY2017]	Key Safety Challenges for the IIoT, IIC Technical White Paper IIC:WHT:IN6:V1.0:PB:20171201, 2017-December-01
	https://www.iiconsortium.org/pdf/Key_Safety_Challenges_for_the_IIoT.pdf
[IIC-JOI9]	The IIC Journal of Innovation, 9th Edition: Trustworthiness. 2018-September.
	www.iiconsortium.org/news/journal-of-innovation-2018-sept
[NIST SP800]	NIST SP800-57—Recommendation for Key Management
	NIST SP800-131A Revision 2—Transitioning the Use of Cryptographic Algorithms and Key Lengths
IEC 62443	IEC 62443-3-3, Section 8, FR-4—Data Confidentiality
	IEC 62443-3-3, Section 7, FR-3—System integrity. Includes concepts of Communications Integrity for data in motion, and Information Integrity for data at rest.
ISO/IEC 19790	ISO/IEC 19790—Information technology -- Security techniques -- Security requirements for cryptographic modules
	https://www.iso.org/standard/52906.html
ISO 7498-2:1989	Security Architecture: General description of security services and related mechanisms covering secure communications between open systems.
	https://www.iso.org/obp/ui/#iso:std:iso:7498:-2:ed-1:v1:en

AUTHORS AND LEGAL NOTICE

A publication of the Security Applicability Task Group, authored by Bassam Zarkout (IGnPower), Niheer Patel (RTI), and Apurva Mohan (Schlumberger).

Acknowledgements: Security Applicability Task Group, co-chaired by Ron Zahavi (Microsoft) and James Clardy (Net Foundry), which is a subgroup of the Security Working Group co-chaired by Sven Schrecker (LHP Engineering Solutions) and Jesus Molina (Waterfall Security Solutions).

Authors: Bassam Zarkout (IGnPower), Apurva Mohan (Schlumberger), Niheer Patel (RTI).

Contributors: The following persons contributed valuable ideas and feedback that significantly improved the content and quality of this document: Simon Rix (Irdeto), Sandy Carielli (Entrust Datacard).

Technical Editor: Stephen Mellor (IIC staff) oversaw the process of organizing the contributions of the Authors and Contributors into an integrated document.