



ISO/IEC TS 30168

Edition 1.0 2024-05

# TECHNICAL SPECIFICATION



---

**Internet of Things (IoT) – Generic trust anchor application programming  
interface for industrial IoT devices**

INTERNATIONAL  
ELECTROTECHNICAL  
COMMISSION

---

ICS 35.020

ISBN 978-2-8322-8518-3

**Warning! Make sure that you obtained this publication from an authorized distributor.**

## CONTENTS

FOREWORD.....	7
INTRODUCTION.....	9
1 Scope.....	10
2 Normative references .....	10
3 Terms and definitions .....	10
4 Abbreviated terms .....	12
5 Architecture .....	14
5.1 General.....	14
5.2 Relation to ISO/IEC 30141 .....	14
5.3 Intended target environment .....	14
5.4 Functional scope.....	15
5.5 Concepts .....	15
5.5.1 Abstraction .....	15
5.5.2 Object information model.....	20
5.5.3 Identifiers .....	22
5.5.4 Personalities.....	23
5.5.5 Profiles .....	24
5.5.6 Device states.....	25
5.5.7 Access control.....	25
5.5.8 Secure element properties.....	26
5.6 Implementation view .....	28
5.6.1 System design considerations .....	28
5.6.2 Personalities.....	29
5.6.3 Profiles .....	30
5.6.4 Device states.....	32
5.6.5 Access control.....	37
5.6.6 GTA API start-up .....	40
6 API specification.....	41
6.1 Overview .....	41
6.2 Language binding .....	46
6.3 Endianness.....	46
6.4 Exception handling .....	46
6.5 Using GTA API from an application.....	46
6.5.1 Header files.....	46
6.5.2 Call conventions and error handling.....	46
6.6 Types and function documentation.....	47
6.6.1 Basic types.....	47
6.6.2 General management functions .....	50
6.6.3 Process synchronization.....	55
6.6.4 Secure memory management .....	59
6.6.5 Function parameter I/O streams .....	61
6.6.6 Instance management functions .....	65
6.6.7 Context management functions.....	67
6.6.8 Access token functions .....	71
6.6.9 Device state management functions .....	75
6.6.10 Identifier and personality management .....	77

6.6.11	Access policy management functions.....	97
6.6.12	Data protection functions .....	106
6.6.13	Channel protection functions .....	109
6.6.14	Supplementary security functions .....	114
6.6.15	Trusted execution environment.....	115
6.6.16	Secure element provider implementation support.....	115
Annex A (normative) GTA API C header files .....		119
A.1	Dependencies .....	119
A.2	Application interface – gta_api.h .....	119
A.3	Provider interface – gta_apif.h .....	119
A.4	Handles – gta_handle.h .....	119
A.5	Function parameter I/O streams – gta_stream.h.....	120
A.6	Error information – gta_errinfo.h .....	120
A.7	Secure memory management – gta_secmen.h .....	120
A.8	Process synchronization – gta_psync.h.....	120
Annex B (normative) Basic profiles .....		121
B.1	ch.iec.30168.basic.passcode .....	121
B.1.1	Description .....	121
B.1.2	Deployment .....	121
B.1.3	Usage.....	122
B.2	ch.iec.30168.basic.local_data_integrity_only .....	122
B.2.1	Description .....	122
B.2.2	Creation .....	122
B.2.3	Usage.....	123
B.3	ch.iec.30168.basic.local_data_protection.....	124
B.3.1	Description .....	124
B.3.2	Creation .....	124
B.3.3	Usage.....	124
Annex C (informative) Example security scenarios for Industrial IoT .....		126
C.1	Analysis of example security scenarios for IIoT.....	126
C.1.1	General .....	126
C.1.2	Scenarios for application protocols .....	126
C.1.3	Secure device identities.....	131
C.1.4	Supply-chain and trustworthiness/authenticity of device.....	132
C.1.5	Device integrity protection .....	133
C.1.6	Application security .....	134
C.1.7	Feature licensing .....	136
C.1.8	Device and machine management .....	137
C.1.9	Blockchain/distributed ledger technology .....	140
C.1.10	GTA management.....	141
C.2	Security requirements for security scenarios .....	142
C.2.1	General .....	142
C.2.2	General or nonfunctional requirements .....	142
C.2.3	Functional security requirements overview and description.....	143
C.2.4	Security requirements for OPC UA.....	145
C.2.5	Security requirements for PROFINET security extensions.....	145
C.2.6	Security requirements for secure communication .....	146
C.2.7	Security requirements for secure device identities .....	146
C.2.8	Security requirements for trustworthiness/authenticity of device .....	147

C.2.9	Security requirements for device integrity protection.....	147
C.2.10	Security requirements for application security.....	148
C.2.11	Security requirements for feature licensing.....	148
C.2.12	Security requirements for device management.....	149
C.2.13	Security requirements for blockchain/distributed ledger technology.....	149
C.2.14	Security requirements for GTA management.....	150
Annex D (informative)	Security classes and attestation.....	151
D.1	Security classes/levels.....	151
D.2	Offline validation of security level by organizational means (out-of-band).....	152
D.3	Online validation of security level by attestation (in-band).....	152
D.3.1	General.....	152
D.3.2	Attestation of SE or GTA API runtime for a specific device.....	152
D.3.3	Attestation of personalities and their attributes.....	152
D.3.4	Attestation of a transaction.....	153
D.3.5	Detached attestation.....	153
Annex E (informative)	Examples for further illustration of GTA API concepts.....	154
E.1	Pre-initial device state example for TPM.....	154
E.2	Composing systems from subsystems containing SEs.....	155
E.3	Example deployment of SEs in a composite system design.....	156
Annex F (informative)	Implementation guidance.....	160
F.1	Host platform abstraction.....	160
F.2	Buffer management.....	160
F.3	Signalling and semaphores.....	160
F.4	Coding style.....	161
F.5	Secure coding.....	161
Annex G (informative)	Example code.....	162
G.1	General.....	162
G.2	Using GTA API with <code>ch.iec.30168.basic.local_data_protection</code> .....	162
G.3	Using GTA API with <code>ch.iec.30168.basic.local_data_integrity_only</code> .....	165
G.3.1	General.....	165
G.3.2	Protection with data recovery.....	165
G.3.3	Detached protection.....	165
G.4	Protecting a personality with <code>ch.iec.30168.basic.passcode</code> .....	167
G.5	Example for a simple buffer stream.....	170
G.5.1	<code>myio_bufstream.h</code> .....	170
G.5.2	<code>myio_bufstream.c</code> .....	172
G.6	Secure element provider template.....	174
Bibliography	.....	175
Figure 1	– GTA API environment.....	15
Figure 2	– GTA API modular architecture interfaces.....	16
Figure 3	– Crypto technology driven API design.....	18
Figure 4	– GTA API security service driven API design.....	19
Figure 5	– Multi-application capability.....	19
Figure 6	– Secure element abstraction.....	20
Figure 7	– Object information model (static view).....	21

Figure 8 – Object information model (runtime view) .....	22
Figure 9 – Value creation chain .....	25
Figure 10 – Device state stack .....	33
Figure 11 – Device state transitions .....	33
Figure 12 – Device state stack (push) .....	34
Figure 13 – Device state stack (pop).....	34
Figure 14 – Access token.....	37
Figure 15 – Personality derived access token .....	39
Figure 16 – Access policy composition (BNF) .....	40
Figure 17 – GTA API start-up phases.....	41
Figure 18 – Example gta_personality_enumerate() .....	78
Figure 19 – Example access policy handling by SE provider .....	99
Figure 20 – Channel protection functions .....	110
Figure A.1 – Dependency graph for gta_api.h.....	119
Figure C.1 – Device management .....	138
Figure E.1 – Composing systems from subsystems containing SEs .....	155
Figure E.2 – Example: Robot as a composite system.....	156
Figure E.3 – Example: SEs deployed within composite system .....	157
Figure E.4 – Example: Component device states .....	159
Table 1 – Access control.....	26
Table 2 – Mapping between SE properties and protection goals .....	28
Table 3 – Properties of personality creation profiles.....	31
Table 4 – Properties of personality deployment profiles .....	31
Table 5 – Properties of personality enrollment profiles .....	31
Table 6 – Properties of personality usage profiles .....	32
Table 7 – GTA API function groups .....	42
Table 8 – GTA API feature classes .....	43
Table 9 – GTA API functions per feature class.....	43
Table 10 – Basic profiles .....	50
Table 11 – GTA API functions with access control .....	71
Table B.1 – ch.iec.30168.basic.passcode deployment properties .....	121
Table B.2 – ch.iec.30168.basic.passcode usage properties .....	122
Table B.3 – ch.iec.30168.basic.local_data_integrity_only creation properties.....	123
Table B.4 – ch.iec.30168.basic.local_data_integrity_only usage properties.....	123
Table B.5 – ch.iec.30168.basic.local_data_protection creation properties .....	124
Table B.6 – ch.iec.30168.basic.local_data_protection usage properties .....	125
Table C.1 – Scenarios for OPC UA client and server .....	127
Table C.2 – Security classes for the PROFINET protocol.....	130
Table C.3 – Scenarios for PROFINET security.....	130
Table C.4 – Scenarios for secure communication protocols .....	131
Table C.5 – Scenarios for secure identities .....	132
Table C.6 – Scenarios for device trustworthiness.....	133

Table C.7 – Scenarios for system integrity protection.....	134
Table C.8 – Scenarios for know-how protection .....	135
Table C.9 – Scenarios for feature licensing.....	137
Table C.10 – Scenarios for device management .....	139
Table C.11 – Scenarios for blockchain/distributed ledger technology (DLT) .....	141
Table C.12 – Scenarios for GTA management .....	142
Table C.13 – General or nonfunctional requirements .....	142
Table C.14 – GTA-API functional security requirements.....	143
Table C.15 – Security requirements for OPC UA.....	145
Table C.16 – Security requirements for PROFINET security extensions.....	146
Table C.17 – Security requirements for secure communication .....	146
Table C.18 – Security requirements for secure device identities .....	147
Table C.19 – Security requirements for trustworthiness/authenticity of device.....	147
Table C.20 – Security requirements for device integrity protection .....	148
Table C.21 – Security requirements for application security .....	148
Table C.22 – Security requirements for feature licensing .....	149
Table C.23 – Security requirements for device management.....	149
Table C.24 – Security requirements for blockchain/distributed ledger technology.....	150
Table C.25 – Security requirements for GTA management.....	150
Table D.1 – Example security levels .....	151

# INTERNET OF THINGS (IoT) – GENERIC TRUST ANCHOR APPLICATION PROGRAMMING INTERFACE FOR INDUSTRIAL IoT DEVICES

## FOREWORD

- 1) ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.
- 2) The formal decisions or agreements of IEC and ISO on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC and ISO National bodies.
- 3) IEC and ISO documents have the form of recommendations for international use and are accepted by IEC and ISO National bodies in that sense. While all reasonable efforts are made to ensure that the technical content of IEC and ISO documents is accurate, IEC and ISO cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC and ISO National bodies undertake to apply IEC and ISO documents transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC and ISO document and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC and ISO do not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC and ISO marks of conformity. IEC and ISO are not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this document.
- 7) No liability shall attach to IEC and ISO or their directors, employees, servants or agents including individual experts and members of its technical committees and IEC and ISO National bodies for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this ISO/IEC document or any other IEC and ISO documents.
- 8) Attention is drawn to the Normative references cited in this document. Use of the referenced publications is indispensable for the correct application of this document.
- 9) IEC and ISO draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). IEC and ISO take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, IEC and ISO had not received notice of (a) patent(s), which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at <https://patents.iec.ch> and [www.iso.org/patents](http://www.iso.org/patents). IEC and ISO shall not be held responsible for identifying any or all such patent rights.

ISO/IEC TS 30168 has been prepared by subcommittee 41: Internet of Things and Digital Twin, of IEC technical committee JTC 1: Information technology. It is a Technical Specification.

This document contains attached files in the form of GTA API C header files and a secure element provider template that are cited in Annex A and Clause G.6. These files are intended to be used as a complement and do not form an integral part of the publication.

The text of this Technical Specification is based on the following documents:

Draft	Report on voting
JTC1-SC41/388/DTS	JTC1-SC41/413/RVDTS

Full information on the voting for its approval can be found in the report on voting indicated in the above table.

The language used for the development of this Technical Specification is English.

**IMPORTANT – The "colour inside" logo on the cover page of this document indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.**

## INTRODUCTION

Industrial Internet of Things (IIoT) devices face increasing security requirements. This insight is especially important as more and more devices become connected directly or indirectly to the Internet. It is essential that IIoT devices are prepared to perform secure communications for service interaction, monitoring, and control.

However, security is often still observed as rather complex by implementers and integrators. This perception often results in realization obstacles when the integration and use of security mechanisms and secure elements (SE) is wanted.

This document provides a versatile application programming interface (API) for security to allow a generic integration of SEs into IIoT devices. The API is vendor independent and also independent regarding the SE technology being deployed. This approach simplifies redesign for different SEs and supports software-hardware co-design for security. SEs offering different security properties facilitate the selection of an SE according to the intended use, protection goals, and other boundary conditions. The API aims at achieving high-level abstraction profiles for security services and mechanisms to avoid typical low-level interoperability complexity and implementation failures. Requirements and architectural constraints from IIoT applications shape the final design of the API and its usability.

The resulting API facilitates the security-by-design defined integration of security components within IIoT components on a large scale. The time-to-market for secured devices is accelerated. Stakeholders will benefit from higher security levels being available for lower prices. Application of updates and continuous improvements of security along the lifecycle of products and systems are facilitated.

The following stakeholders and their corresponding interests play a role for the generic trust anchor application programming interface (GTA API) definition:

- Manufacturers and users of industrial equipment  
Scalable use of adequate (hardware-based) security technologies depending on required security, multivendor support, migration strategy, or long-term suitability.
- Software developers  
Increased robustness due to use of a unified API.  
Ease of use for developers without dedicated security expertise.
- Manufacturers of security ICs or ICs offering security functions  
Promote use of hardware-based trust anchor technologies for IIoT devices.
- Conformity Assessment Bodies

# INTERNET OF THINGS (IoT) – GENERIC TRUST ANCHOR APPLICATION PROGRAMMING INTERFACE FOR INDUSTRIAL IoT DEVICES

## 1 Scope

This document specifies a generic application programming interface (API) for the integration of SEs within Industrial Internet of Things (IIoT) devices. It considers needs from industrial usage scenarios and applications. This document also provides guidance for implementation, testing, and conformity validation.

## 2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEEE Std 802-2014, *IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture*

IETF RFC 1035, P. Mockapetris, "Domain names – implementation and specification", November 1987, available at <https://www.rfc-editor.org/info/rfc1035> [viewed 2023-08-29]

IETF RFC 1779, S. Kille, "A String Representation of Distinguished Names", March 1995, available at <https://www.rfc-editor.org/info/rfc1779> [viewed 2023-08-29]

IETF RFC 4122, P. Leach, M. Mealling, and R. Salz, "A Universally Unique Identifier (UUID) URN Namespace", July 2005, available at <https://www.rfc-editor.org/info/rfc4122> [viewed 2023-08-29]

IETF RFC 8446, E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3", August 2018, available at <https://www.rfc-editor.org/info/rfc8446> [viewed 2023-08-29]