



ISO/IEC 14543-5-1

Edition 1.0 2010-02

INTERNATIONAL STANDARD

**Information technology – Home electronic system (HES) architecture –
Part 5-1: Intelligent grouping and resource sharing for Class 2 and Class 3 –
Core protocol**



THIS PUBLICATION IS COPYRIGHT PROTECTED
Copyright © 2010 ISO/IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester.

If you have any questions about ISO/IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

IEC Central Office
3, rue de Varembe
CH-1211 Geneva 20
Switzerland
Email: inmail@iec.ch
Web: www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

- Catalogue of IEC publications: www.iec.ch/searchpub

The IEC on-line Catalogue enables you to search by a variety of criteria (reference number, text, technical committee,...). It also gives information on projects, withdrawn and replaced publications.

- IEC Just Published: www.iec.ch/online_news/justpub

Stay up to date on all new IEC publications. Just Published details twice a month all new publications released. Available on-line and also by email.

- Electropedia: www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 20 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary online.

- Customer Service Centre: www.iec.ch/webstore/custserv

If you wish to give us your feedback on this publication or need further assistance, please visit the Customer Service Centre FAQ or contact us:

Email: csc@iec.ch
Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00



ISO/IEC 14543-5-1

Edition 1.0 2010-02

INTERNATIONAL STANDARD

**Information technology – Home electronic system (HES) architecture –
Part 5-1: Intelligent grouping and resource sharing for Class 2 and Class 3 –
Core protocol**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

PRICE CODE

XF

ICS 35.200

ISBN 2-8318-1076-0

CONTENTS

FOREWORD.....	8
INTRODUCTION.....	9
1 Scope.....	10
2 Normative references	10
3 Terms, definitions and abbreviations	11
3.1 Terms and definitions	11
3.2 Abbreviations	12
4 Conformance.....	13
4.1 IGRS network.....	13
4.2 IGRS devices	13
5 IGRS architecture.....	14
5.1 Overview	14
5.2 IGRS Core Protocol.....	15
5.3 IGRS Application Profile	15
5.4 IGRS Application	16
5.5 IGRS and other standards	17
6 IGRS device interaction model	18
6.1 Overview	18
6.2 Device online	19
6.3 Device (group) discovery.....	19
6.4 Device pipe setup.....	19
6.5 Device group setup and join	20
6.6 Service discovery	20
6.7 Session setup.....	21
6.8 Service invocation	21
6.9 Session termination.....	21
6.10 Device/Service online/offline event subscription	21
6.11 Device/Service online/offline event notification	22
6.12 Device/Service online/offline event unsubscription	22
6.13 Pipe disconnection	22
6.14 Device group dismiss and secession	22
6.15 Device offline	23
7 IGRS message framework	23
7.1 Overview	23
7.2 IGRS request/notification message structure	24
7.3 IGRS response message structure	25
7.4 IGRS message based on secure device pipe	26
7.4.1 Message authentication code generation	26
7.4.2 Message encryption.....	27
8 IGRS device and service description	27
8.1 IGRS device description	27
8.1.1 Device description template.....	27
8.1.2 Device identifier.....	31
8.1.3 Device group identifier.....	31
8.1.4 Device type identifier	32
8.1.5 Device security mechanism descriptor	33

8.2	IGRS service description	40
8.2.1	Service description template	40
8.2.2	Service identifier	51
8.2.3	Service type identifier	51
8.2.4	Service access control	52
8.2.5	Identity authentication mechanism of service access control	53
8.3	IGRS client description	56
8.4	IGRS user description	56
9	IGRS device grouping	56
9.1	Device advertisement	56
9.1.1	Device online advertisement	56
9.1.2	Device offline advertisement	58
9.2	Device pipe management	58
9.2.1	Unsecure device pipe setup	58
9.2.2	Unsecure device pipe maintenance	58
9.2.3	Secure device pipe setup	59
9.2.4	Secure device pipe teardown	79
9.2.5	Device trust relationship	80
9.2.6	Device online detection	80
9.3	Detailed device description document retrieval	81
9.3.1	Retrieve detailed device description document request	81
9.3.2	Retrieve detailed device description document response	82
9.4	Retrieve detailed device description document based on non-secure pipe	84
9.5	Device group setup	84
9.5.1	Global peer-to-peer device group	84
9.5.2	Specified peer-to-peer device group	84
9.5.3	Centralised device group	86
9.6	Device search	90
9.6.1	Device search based on multicast	90
9.6.2	Device search by proxy	94
9.7	Device online/offline event subscription	97
9.7.1	Device online/offline event subscription request	97
9.7.2	Device online/offline event subscription renewal request	99
9.7.3	Device online/offline event subscription response	100
9.7.4	Device online/offline event unsubscription	101
9.7.5	Device online/offline event notification	102
9.8	Device group search	104
9.8.1	Device group search request message	104
9.8.2	Device group search response message	105
10	IGRS resource sharing	107
10.1	Service online advertisement	107
10.1.1	Service online advertisement based on multicast	107
10.1.2	Service online registration and offline notification based on device pipe	109
10.2	Service search	111
10.2.1	Service search based on multicast	111
10.2.2	Service search by proxy	114
10.3	Service online/offline event subscription	117
10.3.1	Service online/offline event subscription request	117

10.3.2	Service online/offline event subscription renewal request	119
10.3.3	Service online/offline event subscription response	120
10.3.4	Service online/offline event unsubscription	121
10.3.5	Service online/offline event notification	122
10.4	Service description document retrieval	124
10.4.1	Retrieve service description document request	124
10.4.2	Retrieve service description document response	126
10.4.3	Other approaches to retrieve service description documents	127
10.5	Session	127
10.5.1	Session setup condition	127
10.5.2	Common session setup and teardown process	127
10.5.3	Session setup when service access control in master/slave device group is not consistent with device pipe security attribute	131
10.6	Service invocation	137
10.6.1	Service invocation request message	137
10.6.2	Service invocation response message	137
10.6.3	Notification message based on session	138
11	Request/response status codes	139
Annex A	(normative) IGRS service discovery protocols (ISDP)	142
A.1	General	142
A.2	ISDP message format	142
A.2.1	General	142
A.2.2	ISDP start-lines	142
A.2.3	ISDP message headers	142
A.2.4	ISDP processing rules	143
A.3	ISDP usage in IGRS specification	143
Annex B	(normative) Description documents	145
B.1	Specification description	145
B.2	Session description	249
B.3	Service description	250
B.4	Pipe description	253
B.5	Device template	254
B.6	Master slave device group advertisement	256
B.7	Device type list	256
B.8	Peer-to-peer device group advertisement	257
B.9	Device description	257
Bibliography		259

Figure 1 – IGRS specification framework	15
Figure 2 – IGRS application interaction	17
Figure 3 – IGRS device interaction model	18
Figure 4 – Secure device pipe setup	59
Table 1 – IGRS request and notification message	24
Table 2 – IGRS response message	25
Table 3 – Message authentication code	27
Table 4 – Device security mechanism protocol algorithm	40
Table 5 – Service access control policy	53
Table 6 – Device authentication mechanisms and the corresponding encryption algorithm descriptor	56
Table 7 – Device online advertisement	57
Table 8 – Device offline advertisement	58
Table 9 – Pipe setup request based on symmetric-key cryptosystem	60
Table 10 – Pipe setup response based on symmetric-key cryptosystem	61
Table 11 – Pipe setup request based on symmetric-key authentication, encrypted message transmission, and authentication mechanism	61
Table 12 – Pipe setup response based on symmetric-key authentication, encrypted message transmission, and authentication mechanism	62
Table 13 – Pipe setup request based on authentication, encrypted message transmission, and authentication mechanism of public-key cryptosystem	63
Table 14 – Pipe setup response based on authentication, encrypted message transmission, and authentication mechanism of public-key cryptosystem	63
Table 15 – Pipe setup request based on trusted third party authentication, encrypted message transmission, and authentication mechanism	64
Table 16 – Pipe setup response based on trusted third party authentication, encrypted message transmission, and authentication mechanism	64
Table 17 – Authentication request based on identity authentication and message authentication mechanism of symmetric-key cryptosystem	65
Table 18 – Authentication response based on identity authentication and message authentication mechanism of symmetric-key cryptosystem	66
Table 19 – Authentication result request based on identity authentication and message authentication mechanism of symmetric-key cryptosystem	67
Table 20 – Authentication result response based on identity authentication and message authentication mechanism of symmetric-key cryptosystem	67
Table 21 – Authentication request based on identity authentication and encrypted message transmission and authentication mechanism of symmetric-key cryptosystem	68
Table 22 – Authentication response based on identity authentication and encrypted message transmission and authentication mechanism of symmetric-key cryptosystem	69
Table 23 – Authentication result request based on identity authentication and encrypted message transmission and authentication mechanism of symmetric-key cryptosystem	70
Table 24 – Authentication result response based on identity authentication and encrypted message transmission and authentication mechanism of symmetric-key cryptosystem	70
Table 25 – Authentication request based on authentication and encrypted message transmission and authentication mechanism of public-key cryptosystem	71

Table 26 – Authentication response based on authentication and encrypted message transmission and authentication mechanism of public-key cryptosystem	72
Table 27 – Authentication result request based on authentication and encrypted message transmission and authentication mechanism of public-key cryptosystem	73
Table 28 – Authentication result response based on authentication and encrypted message transmission and authentication mechanism of public-key cryptosystem	74
Table 29 – Authentication request based on authentication, encrypted message transmission, and authentication mechanism of trusted third party	75
Table 30 – Authentication response based on authentication, encrypted message transmission, and authentication mechanism of trusted third party	76
Table 31 – Authentication result request based on authentication, encrypted message transmission, and authentication mechanism of trusted third party	77
Table 32 – Authentication result response based on authentication, encrypted message transmission, and authentication mechanism of trusted third party	78
Table 33 – Secure device pipe setup confirmation request	78
Table 34 – Secure device pipe setup confirmation response	79
Table 35 – Secure device pipe teardown notification message	80
Table 36 – Trust relationship formed between devices after pipe setup	80
Table 37 – Device online detection request message	81
Table 38 – Device online detection response message	81
Table 39 – Device description document retrieval request message	82
Table 40 – Device description document retrieval response message	83
Table 41 – Device group advertisement message of specified peer-to-peer device group	85
Table 42 – Device leaves a specified peer-to-peer device group quit group message	86
Table 43 – Device group advertisement message of master-slave device group	87
Table 44 – Request message to join a master-slave device group	88
Table 45 – Response message to join a master-slave device group	88
Table 46 – Device group dissolve notification message sent by master device	89
Table 47 – Withdraw notification message sent by slave device	89
Table 48 – Device search request message	90
Table 49 – Device search response message	92
Table 50 – Device search request that slave device generates to master device	94
Table 51 – Device search response message	96
Table 52 – Device event subscription request message	98
Table 53 – Device online/offline event subscription renewal request message	100
Table 54 – Device online/offline event subscription response message	101
Table 55 – Device online/offline event unsubscription message	102
Table 56 – Device online/offline event notification message	103
Table 57 – Device group search request message	105
Table 58 – Device group search response message	106
Table 59 – Service online advertisement message	108
Table 60 – Service offline advertisement message	109
Table 61 – Service online registration notification message	110
Table 62 – Service offline notification message	111
Table 63 – Multicast-based service search request message	112



Table 64 – UDP unicast-based service search request message.....	113
Table 65 – Service search request message by proxy.....	115
Table 66 – Service search response message by proxy	116
Table 67 – Service online/offline event subscription request message	118
Table 68 – Service online/offline event subscription renewal request	120
Table 69 – Service online/offline event subscription response.....	121
Table 70 – Service online/offline event unsubscription message	122
Table 71 – Service online/offline event notification message.....	123
Table 72 – Retrieve service description document request message.....	125
Table 73 – Retrieve service description document response message	126
Table 74 – Common session setup request message.....	128
Table 75 – Token setup and structure	129
Table 76 – Common session setup response message	130
Table 77 – Common session teardown notification message.....	131
Table 78 – Retrieve session encryption key generation request	133
Table 79 – Retrieve session encryption key generation response	134
Table 80 – Session encryption key transfer request message	135
Table 81 – Session encryption key transfer response message.....	136
Table 82 – Service invocation request message.....	137
Table 83 – Service invocation response message	138
Table 84 – Notification message based on session	139
Table 85 – Response status code category.....	140
Table 86 – Response status code definition	141

INFORMATION TECHNOLOGY – HOME ELECTRONIC SYSTEM (HES) ARCHITECTURE –

Part 5-1: Intelligent grouping and resource sharing for Class 2 and Class 3 – Core protocol

FOREWORD

- 1) ISO (International Organization for Standardization) and IEC (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. Their preparation is entrusted to technical committees; any ISO and IEC member body interested in the subject dealt with may participate in this preparatory work. International governmental and non-governmental organizations liaising with ISO and IEC also participate in this preparation.
- 2) In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.
- 3) 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 member bodies.
- 4) IEC, ISO and ISO/IEC publications have the form of recommendations for international use and are accepted by IEC and ISO member bodies in that sense. While all reasonable efforts are made to ensure that the technical content of IEC, ISO and ISO/IEC publications is accurate, IEC or ISO cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 5) In order to promote international uniformity, IEC and ISO member bodies undertake to apply IEC, ISO and ISO/IEC publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any ISO/IEC publication and the corresponding national or regional publication should be clearly indicated in the latter.
- 6) ISO and IEC provide no marking procedure to indicate their approval and cannot be rendered responsible for any equipment declared to be in conformity with an ISO/IEC publication.
- 7) All users should ensure that they have the latest edition of this publication.
- 8) No liability shall attach to IEC or ISO or its directors, employees, servants or agents including individual experts and members of their technical committees and IEC or ISO member 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 of, use of, or reliance upon, this ISO/IEC publication or any other IEC, ISO or ISO/IEC publications.
- 9) Attention is drawn to the normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 10) Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 14543-5-1 was prepared by subcommittee 25: Interconnection of information technology equipment, of ISO/IEC joint technical committee 1: Information technology.

The list of all currently available parts of the ISO/IEC 14543 series, under the general title *Information technology – Home electronic system (HES) architecture*, can be found on the IEC web site.

This International Standard has been approved by vote of the member bodies, and the voting results may be obtained from the address given on the second title page.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

INTRODUCTION

ISO/IEC 14543-5, Intelligent Grouping and Resource Sharing for HES (IGRS), is divided into six parts:

➤ **IGRS Part 5-1: Core Protocol**

- Specifies the TCP/IP protocol stack as the basis and the HTTP protocol as the message-exchanging framework among devices.
- Defines a series of device and service interaction/invoke standards, including device and service discovery protocol, device and service description, service invocation, security mechanisms, etc.
- Specifies core protocols for a type of home network that supports streaming media and other high-speed data transport within a home.

➤ **IGRS Parts 5-2#: Application profile** (under consideration)

- Based on the IGRS Core Protocol.
- Defines a device and service interaction mechanism, as well as application interfaces used in IGRS Basic Applications.
- Multiple application profiles are being developed, including:
 - Part 5-21: AV Profile (under consideration)
 - Part 5-22: File Profile (under consideration)
- Additional application profiles are planned (part numbers to be assigned; these projects are under consideration)
 - Part 5-2w: DVD Profile
 - Part 5-2x: QoS Profile
 - Part 5-2y: DMCP Profile
 - Part 5-2z: Universal Control Profile

➤ **IGRS Part 5-3: Basic Application** (under consideration)

- Includes an IGRS basic application list.
- Defines a basic application framework.
- Addresses operation specifics (device grouping, service description template, etc.), function definitions, and service invocation interfaces.

➤ **IGRS Part 5-4: Device Validation** (under preparation)

- Defines a standard method to validate an IGRS-compliant device.

➤ **IGRS Part 5-5: Device Types** (under consideration)

- Defines IGRS Device types used in IGRS applications.

➤ **IGRS Part 5-6: Service Types** (under consideration)

- Defines basic service types used in IGRS applications.

INFORMATION TECHNOLOGY – HOME ELECTRONIC SYSTEM (HES) ARCHITECTURE –

Part 5-1: Intelligent grouping and resource sharing for Class 2 and Class 3 – Core protocol

1 Scope

This part of the ISO/IEC 14543 specifies the services and protocol of the application layer for use by IGRS Devices in the Home Electronic System. An IGRS Device (Intelligent Grouping and Resource Sharing Device) includes the communications protocol specified in the multiple parts of ISO/IEC 14543-5. The objective of this standard is to enable resource sharing and service collaboration among devices. This standard describes:

- the interoperability mechanism;
- the process and messaging format of device discovery and device grouping;
- the process and messaging format of resource sharing among IGRS Devices;
- IGRS Device and service description requirements.

This standard is applicable to resource sharing and service collaboration among computers, consumer electronics, and communication devices in a Local Area Network (LAN) or Personal Area Network (PAN) environment, especially in a wireless dynamic network.

2 Normative references

The following referenced documents are indispensable for the application 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.

The provisions of the referenced specifications other than ISO/IEC, IEC, ISO and ITU documents, as identified in this clause, are valid within the context of this International Standard. The reference to such a specification within this International Standard does not give it any further status within ISO or IEC. In particular, it does not give the referenced specification the status of an International Standard.

ISO/IEC 9594-8:2005, *Information technology – Open Systems Interconnection – The directory: Public-key and attribute certificate frameworks*

ISO/IEC 10118-3:2004, *Information technology – Security techniques – Hash-functions – Part 3: Dedicated hash-functions*

ISO/IEC 18033-3, *Information technology – Security techniques – Encryption algorithms – Part 3: Block ciphers*

ISO/IEC 19790, *Information technology – Security techniques – Security requirements for cryptographic modules*

ISO/IEC 29341-1:2008, *Information technology – UPnP Device Architecture – Part 1: UPnP Device Architecture Version 1.0*

IEEE 1363:2000, *Standard Specifications For Public Key Cryptography*

IETF RFC 1510: *The Kerberos Network Authentication Service (V5)*

IETF RFC 1766: *Tags for the Identification of Languages*

IETF RFC 2234: *Augmented BNF for Syntax Specifications: ABNF*

IETF RFC 2616: *Hypertext Transfer Protocol -- HTTP/1.1*

IETF RFC 2774: *An HTTP Extension Framework*

IETF RFC 3447: *Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1*

W3C-REC-XML-1998-210:1998, *Extensible Markup Language (XML) 1.0*

W3C SOAP 1.2: *Simple Object Access Protocol Version 1.2*

<http://www.w3.org/2002/12/soap-envelope>

W3C WSDL 2.0: *Web Service Description Language Version 2.0*

<http://www.w3.org/TR/wsd120/>

3 Terms, definitions and abbreviations

3.1 Terms and definitions

For the purposes of this document the following terms and definitions apply.

3.1.1

centralised device group

set of IGRS Devices with one IGRS Device acting as the master

NOTE 1 The master is responsible for managing the setup, for dismissing a Device Group, and for processing a join request from other devices.

NOTE 2 The master device and other IGRS Devices in such a Device Group form a centralised or master-slave relationship.

3.1.2

client identifier

unique identifier associated with a Client on an IGRS Device to which that Client belongs

3.1.3

device group

multiple IGRS Devices that are organised into a logical group through the device group management mechanism in the IGRS specification

NOTE Each IGRS Device in a Device Group follows common interaction rules. Two types of Device Groups are defined: peer-to-peer Device Group and centralised (master-slave) Device Group.

3.1.4

device identifier

globally unique device identifier associated with one IGRS Device

3.1.5

device pipe

channel used to transfer device interaction messages

NOTE This channel is set up through the pipe setup mechanism in the IGRS Specification.

3.1.6

IGRS client

application that draws upon the services of one or more connected IGRS Devices

NOTE Multiple client instances can exist on a network at the same time.

3.1.7

IGRS device

information device that conforms to the IGRS specification

3.1.8

IGRS service

sharable resource encapsulated in an IGRS Device by implementing application interfaces and providing services for other IGRS Devices

NOTE An IGRS Service has an invocation interface that meets the requirements of the IGRS specification. These invocation interfaces are described and announced on the network through the IGRS Service Description Specification.

3.1.9

IGRS specification

ISO/IEC 14543-5 series of standards

3.1.10

IGRS user

owner of an IGRS Device and Client

3.1.11

peer-to-peer device group

set of IGRS Devices where each IGRS Device in this set has a peer-to-peer relationship with each other

3.1.12

service identifier

unique identifier assigned to a service provided by a specific IGRS Device

NOTE Note that the same type of service may be provided by multiple IGRS Devices within the same network. Each instance of a service has a unique service identifier on the IGRS Device providing that service.

3.1.13

service type

category of IGRS Service defined according to the set of resources encapsulated

NOTE The Service Type enables service applications in the same category to have common invocation interfaces.

3.1.14

user identifier

identifier of an IGRS user

3.2 Abbreviations

The following acronyms and abbreviations are used in this standard and commonly used in other industry publications.

3DES	3 Data Encryption Standard
ABNF	Augmented Backus-Naur Form
AES	Advanced Encryption Standard
CA	Certificate Authority

ECC-192	192-bit Elliptic Curve Cryptography
ElGamal	ElGamal Asymmetric Key Encryption Algorithm (Digital Signature Algorithm), see IEEE 1363-2000
FIPS	Federal Information Processing Standards
HTTP	Hypertext Transport Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IGRS	Intelligent Grouping and Resource Sharing, see ISO/IEC 14543-5 series
IGRS/1.0	Version 1.0 of the IGRS protocol, see this International Standard
ISDP	IGRS Service Discovery Protocol, see this International Standard
IP	Internet Protocol
PRNG	Pseudo-random Number Generator
RNG	Random Number Generator
RSA-1024	Rivest-Shamir-Adleman-1024
SHA-1	Secure Hash Algorithm 1
SHA-256	Secure Hash Algorithm 256
SSDP	Simple Service Discovery Protocol
SOAP	Simple Object Access Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
URL	Uniform Resource Locator
URI	Uniform Resource Identifier
UPnP	Universal Plug and Play
UUID	Universally Unique Identifier
W3C	World Wide Web Consortium
WSDL	Web Service Description Language
XML	Extended Markup Language, see W3C-REC-XML-1998-210:1998

4 Conformance

4.1 IGRS network

An IGRS Network is defined by the architecture specified in Clause 5, the IGRS Device interaction model specified in Clause 6, the message formats specified in Clause 7, device grouping procedures specified in Clause 9, and the resource sharing model specified in Clause 10. All mandatory requirements shall be implemented for conformance with this standard.

4.2 IGRS devices

To be considered IGRS compliant, each device shall meet the criteria defined by the IGRS Device interaction model requirements specified in Clause 6, the message format (sent and received) requirements specified in Clause 7, and the device and service description requirements according to the templates specified in Clause 8.

In addition, a basic IGRS-compliant device shall also support the following requirements.

- The device shall be able to send a Device Online Advertisement Message as specified in 9.1.

- The device shall support other IGRS Devices to set up secure pipes based on the null security mechanism using a secure Device Pipe IP address specified in an online advertisement message.
- The device shall be able to respond as specified in 9.3 to Device Description Document Retrieval Request Messages from other IGRS Devices.
- The device shall be able to respond as specified in 9.6 to Device Search Request Messages from other IGRS Devices.
- The service on the device shall be able to send Service Online/Offline Advertisement Messages as specified in 10.1.
- The device shall be able to respond as specified in 10.2 to Device Search Request Messages Based on Multicast from other IGRS Devices.
- The device shall be able to respond as specified in 10.4 to Service Description Document Retrieval Request Messages from other IGRS Devices.
- The device shall be able to respond as specified in 10.5 to Common Session Setup Request Messages from other IGRS Device, and establish session accordingly.
- The device shall be able to respond as specified in 10.6 to Service Invocation Request Messages from other IGRS Devices.

5 IGRS architecture

5.1 Overview

The scope of the ISO/IEC 14543-5 series of standards, called the IGRS specification includes enterprises, public areas, and homes. The IGRS specification includes a universal resource description and a universal service interface. The IGRS specification was developed for the following purposes:

- to support device resource sharing and synergistic applications among diversified information devices, consumer electronic devices, and communication devices in a LAN or PAN network;
- to enable the interoperability among the devices and to make them easy to use, utilising the features and functionality of each device.

The IGRS Specification Architecture is shown in Figure 1. It is based on the TCP/IP protocol suite (specified in IETF RFC 793) and uses the HTTP protocol as the framework. The IGRS Core Protocol supports wired LAN, ISO/IEC 8802-3-family network media, and ISO/IEC 8802-11, as well as Bluetooth. IGRS defines device interaction specifications based on the HTTP/1.1 protocol, including the IGRS Core Protocol comprising the platform of device discovery and resource sharing, and the IGRS Application Profile comprising the collaborative service platform.

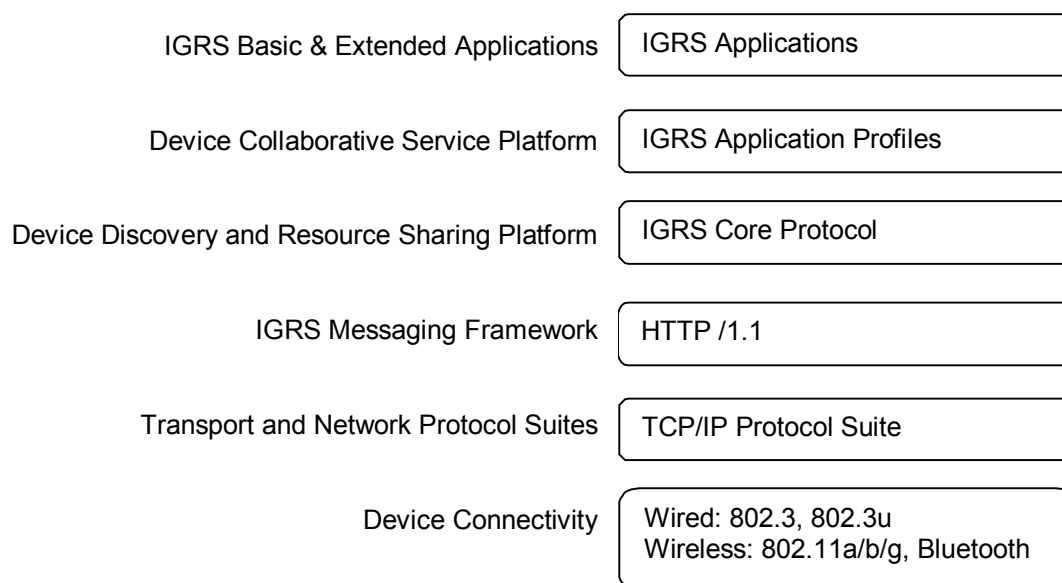


Figure 1 – IGRS specification framework

The techniques specified in the IGRS specification for device connection, data transfer, network protocol, and device message interaction framework are all widely-adopted, mature, open industry technologies. As more and more new network technologies become mature, IGRS may be amended to support these technologies (such as IPv6, etc.). ISO/IEC 14543-5-1 to ISO/IEC 14543-5-6 specify three components: IGRS Core Protocol, IGRS Application Profile, and IGRS Basic Application. The IGRS Core Protocol defines IGRS Device grouping and the interaction mechanism between client and service. Based on the IGRS Core Protocol, the IGRS Application Profiles define service descriptions and interaction logic for various IGRS applications. The various IGRS applications are standardised and shall be implemented as specified in the corresponding IGRS Application Profiles to ensure interoperability. Additional functions may be implemented using protocols not specified in the ISO/IEC 14543-5 series. An application combining a profile specified in ISO/IEC 14543-5-2# with private functions is called an IGRS extended application. The development of an IGRS extended application is specified in ISO/IEC 14543-5-3.

5.2 IGRS Core Protocol

The IGRS Core Protocol, specified in ISO/IEC 14543-5-1, defines the mechanism of IGRS Device discovery and resource sharing, including a device discovery mechanism, Device Pipe setup mechanism, service discovery mechanism, device group management mechanism, session management mechanism, service access mechanism, etc.

The device discovery mechanism defines the process by which an IGRS Device advertises itself and discovers other device information on the network. The Device Pipe setup mechanism defines the process by which two devices that have discovered each other establish a reliable connection pipe. The device group management mechanism defines the rules by which multiple devices form a specific device group on the basis of the connection pipe. The service discovery mechanism defines the process by which one device discovers the service information announced by other devices in the same device group. The session mechanism defines the process of session setup and management for supporting client and service access. The service access mechanism defines the rules that shall be followed in order to complete service invocation between the IGRS Client and the IGRS Service.

5.3 IGRS Application Profile

An IGRS Application Profile, specified in multiple parts of ISO/IEC 14543-5-2x series, is a series of application interaction rules defined for end applications based on the IGRS Core

Protocol. For example, the audio/video framework of home multimedia defines the IGRS Services that every audio/video device shall implement for applications in a home environment, and the interaction rules between the service provider and the client.

5.4 IGRS Application

IGRS applications, specified in ISO/IEC 14543-5-3, are based on the IGRS Core Protocol, specified in ISO/IEC 14543-5-1 and an IGRS Application Profile, specified in ISO/IEC 14543-5-2#. IGRS applications include two categories: IGRS basic applications that are fully specified in ISO/IEC 14543-5-3 and IGRS extended applications (that include vendor specific extensions). An IGRS extended application is based on the IGRS basic protocol and IGRS application profile. An IGRS extended application conforms to the IGRS specification and is intended to diversify and to enhance the current IGRS device functionalities. It is to be noted that IGRS Devices are categorised into different types based on different usage scenarios. IGRS basic applications are standardised applications that relate to specific device types and possess the basic functionalities of an IGRS Device, as defined in ISO/IEC 14543-5-5. In addition to basic applications implemented by IGRS Devices, application developers have the freedom to develop various applications conforming to the IGRS specifications. The developers shall implement the IGRS Core Protocol and may extend an IGRS Application Profile with proprietary functions to fully utilise the functionalities provided by their IGRS Devices. This kind of application extension is called an IGRS extended application.

An IGRS application is implemented through the interaction between one or more services and one or more clients that use these services. A typical IGRS application interaction is shown in Figure 2.

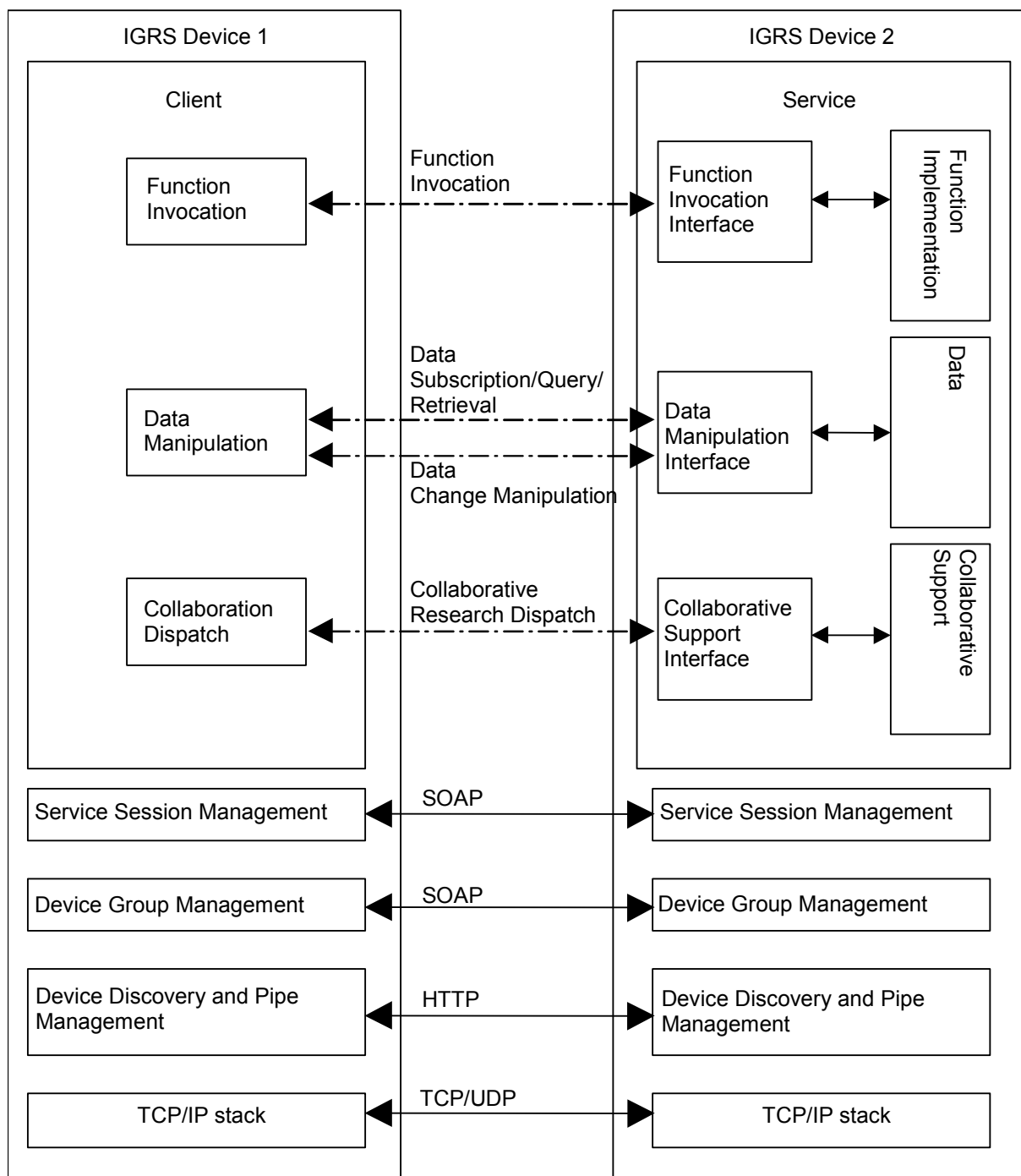


Figure 2 – IGRS application interaction

5.5 IGRS and other standards

The IGRS specification is in part based on existing specifications. The IGRS specification uses the HTTP/1.1 and SOAP/1.2 protocols (specified in <http://www.w3.org/2002/12/soap-envelope>) as the message framework for device interaction. IGRS supports interoperability with many other specifications due to the wide adoption of the HTTP/1.1 and SOAP/1.2 protocols. Therefore, the IGRS specification provides a foundation for interoperability.

6 IGRS device interaction model

6.1 Overview

The IGRS Device Interaction Model shown in Figure 3 describes the typical process of an IGRS Device coming online, discovering other IGRS Devices, joining an IGRS Device group, discovering the services of other IGRS Devices in the device group, performing service invocation, and finally going offline. It includes 14 steps (see 6.2 through 6.15).

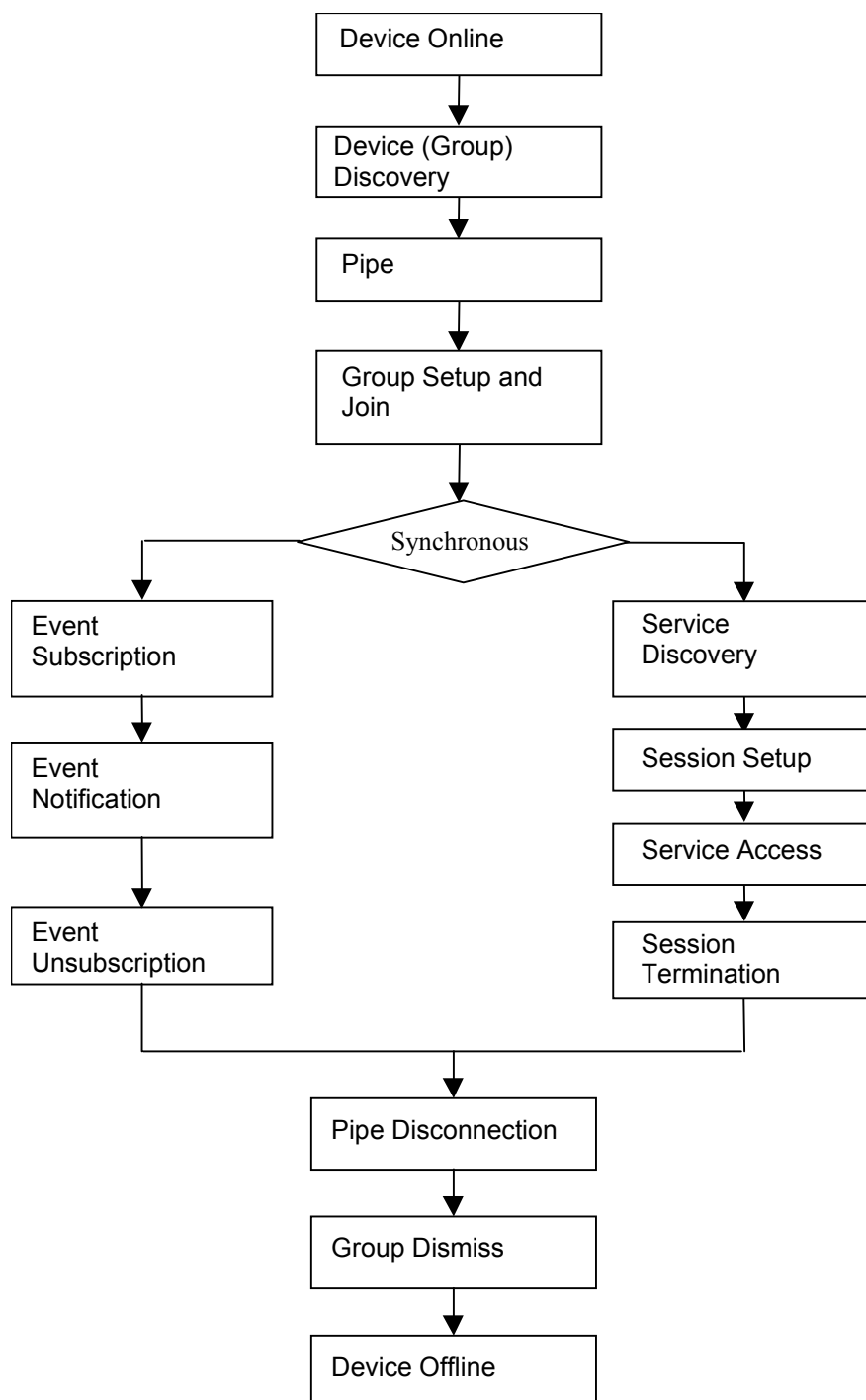


Figure 3 – IGRS device interaction model

6.2 Device online

An IGRS Device shall send a device online advertisement to the specified multicast addresses on all network connections on a regular time interval. The device advertisement shall contain such information as device name, device model, device security attribute requirement, device detailed information access address, etc. (see 9.1.1).

6.3 Device (group) discovery

An IGRS Device may listen to the device online advertisement information on the specified multicast addresses to discover an IGRS Device (group) existing on the network.

An IGRS Device may send a device (group) search requests to the specified multicast address on all networks to which it connects. The IGRS Device (group) that receives this request and matches the search criteria shall send the unicast device search response to the requesting IGRS Device (see 9.6 and 9.8).

6.4 Device pipe setup

Except for an UDP-based multicast search (specified in IETF RFC 768) and unicast response interaction among IGRS Devices, which do not require the setup of a Device Pipe, all the other interactions among IGRS Devices shall be based on the setup of a Device Pipe.

After discovering the target IGRS Device on the network, an IGRS Device may establish a secure pipe with the target IGRS Device through the Device Pipe Management mechanism (see 9.2).

Through the Device Pipe setup mechanism, the IGRS specification encapsulates and simplifies the TCP connection setup and management process during IGRS application interaction. Only one Device Pipe exists between two IGRS Devices at the same time.

Each IGRS Device should control the pipe setup requests from other IGRS Devices by establishing an authorised/accessible IGRS Device list and the maximum number of concurrent pipes.

Device pipes are divided into two categories.

- Secure Device Pipe: the Device Pipe setup is based on the security mechanism commonly supported by the two devices, which is authenticated by this mechanism.
- Unsecure Device Pipe: the Device Pipe is setup without the security mechanism commonly supported by the two devices (not authenticated).

Each IGRS Device online advertisement message shall contain information about the security mechanism and relevant parameters, which are used in the process of Device Pipe setup. An IGRS Device may resolve the target IGRS Device online advertisement message in order to select an appropriate security mechanism to establish a secure Device Pipe with the target IGRS Device.

The IGRS Device pipe mechanism includes Device Pipe setup and Device Pipe management. Two IGRS Devices shall determine the mutual trust relationship through the process of secure Device Pipe setup.

After the Device Pipe setup is completed, two IGRS Devices may detect the online status of each other interactively by exchanging device online-status detection request and response messages.

6.5 Device group setup and join

IGRS Device groups are further divided into three categories: global peer-to-peer device group, specified peer-to-peer device group, and centralised device group.

After coming online, an IGRS Device is placed in a global peer-to-peer device group by default. All IGRS Devices belong to the same global peer-to-peer device group (see 9.5).

An IGRS Device may join or set up a peer-to-peer device group through device configuration. The security interaction rules of all IGRS Devices in the same peer-to-peer device group shall be determined upon negotiation (outside of this IGRS specification) through the secure pipe (see 9.5.2). The master device shall control the joining of other devices through setting the accessible/authorised device list and the maximum number of devices in the device group.

The user may set up a centralised device group and designate a device as the master of this device group. The master device is responsible for periodically sending centralised device group advertisement messages by multicast (see 9.5.3).

After discovering the existence of a centralised device group in the network, an IGRS Device may initiate a joining request to the master device of this centralised device group. The master device shall decide whether to approve the request according to relevant rules. The application shall define specific grouping requirements based on needs, for example, setting up association according to service and device types, etc. IGRS framework supports the establishment of device groups that provide certain sets of services or consist of similar device capabilities on similar physical devices.

An IGRS Device may setup and/or join a specified peer-to-peer device group and a centralised device group.

An IGRS Device may setup and/or join many specified peer-to-peer device groups and centralised device groups at the same time. The creation of each device group: global peer-to-peer device group, specified peer-to-peer device group and centralised device group shall be determined by the requirements indicated in specific guidelines as defined in IGRS basic applications.

6.6 Service discovery

An IGRS Client may listen to the assigned multicast address for the service online advertisement message to discover an existing IGRS Service on the network.

An IGRS Client may initiate the request for a detailed device description document of the target device. The detailed device description document includes the device service list, which describes for all services the names, types, and the URL addresses that point to detailed service description documents (see 10.4).

An IGRS client may send a service search request via a specified multicast address to all IGRS Devices in the network. The search message should contain the type, name, and other attributes of the services. The IGRS Device that receives the service search request and contains the service matching the search criteria shall send an appropriate service search response message to the IGRS Client (see 10.2).

An IGRS Client may send a service search request to a specified IGRS Device through a Device Pipe. The IGRS Device that receives this search request shall return the corresponding search results according to the search conditions (see 10.2).

An IGRS Client in a centralised device group may send a service online/offline event subscription request to the specified IGRS Device through a Device Pipe. The subscription criteria shall be described in the subscription request (see 10.3).

6.7 Session setup

After discovering a service on the target IGRS Device through the service discovery mechanism, an IGRS Client may establish a supporting environment for subsequent service access through the session setup mechanism based on a Device Pipe (see 10.5).

An IGRS Service may attain service access control by gathering the authorised user and device access list of this service. IGRS Services should set up the maximum number of concurrent access.

IGRS Device relationships are determined by whether devices are located in the same device group, whether the IGRS Client device is the trusted device of the IGRS Service device, and whether the IGRS Client device is the specified trusted device of the IGRS Service device.

During the session setup process, the IGRS Device providing an IGRS Service shall acquire information about the IGRS Client device, such as the user ID and client authentication information. Based on this information, the IGRS Client access right shall be authenticated. Meanwhile, concurrent control shall be realised according to the concurrent support capabilities of IGRS Services. Only one session shall exist between one IGRS Client and one IGRS Service at the same time. After a session is successfully established, an IGRS Client may invoke IGRS Services including function invocation, data query, data retrieval, etc. through invoking the interfaces of the IGRS Services.

6.8 Service invocation

After establishing a session with an IGRS Service, an IGRS Client may invoke IGRS Services through pre-defined service invocation mechanisms, in accordance with the interfaces described in the IGRS Service description document (see 10.6).

6.9 Session termination

When an IGRS Client finishes invoking IGRS Services, the IGRS Client shall terminate the session with the IGRS Service (see 10.6). An IGRS Service may initiate the termination of a session with an IGRS Client.

6.10 Device/Service online/offline event subscription

After two IGRS Devices have set up a Device Pipe, the IGRS Client may initiate a device/service online/offline event subscription request with the other party. The IGRS Device that receives the subscription request shall determine whether to accept the subscription; if it accepts the subscription, it then sends the result back to the IGRS Device containing the IGRS Client using an event subscription response message.

An IGRS Client may initiate an event subscription request to the target IGRS Device in the same peer-to-peer device group to subscribe to the service status change on the target device, including service online/offline.

An IGRS Client may initiate an event subscription request to the master device in a centralised device group to subscribe to events in the device group, including device online/offline and service online/offline.

The valid period of event subscription shall be determined by the IGRS Device that accepts the event subscription request. The requesting IGRS Client shall be notified in the subscription response message. An IGRS Client may initiate a subscription renewal request during the valid period. If the IGRS Client does not send the subscription renewal request during the valid period of event subscription, the IGRS Device that accepted the event subscription shall delete all relevant subscription records.

6.11 Device/Service online/offline event notification

When detecting the occurrence of events matching the subscription conditions, the IGRS Device that accepted the event subscription request shall send an event notification message to the IGRS Client through a Device Pipe. The content of the events shall be described in the message.

6.12 Device/Service online/offline event unsubscription

After successfully achieving event subscription with a target IGRS Device, an IGRS Client may send an unsubscription request to the target IGRS Device through a Device Pipe within the valid period of the event subscription.

6.13 Pipe disconnection

After a Device Pipe has been successfully established between two IGRS Devices, the pipe may be disabled under the following conditions.

- a) A device goes offline: after the Device Pipe has been established, the pipe between the two devices shall be disabled immediately when the IGRS Device at either end detects the other end has gone offline.
- b) Pipe idle times out: after the Device Pipe is established, if no session-based message on the pipe exists and no device interaction occurs in a specific time interval except for a detection message of online/offline device status, the IGRS Device at either end shall disable the pipe.
- c) One end initiated disconnection: after the Device Pipe is established, the device at either end may send a pipe disconnection notification to the device at the other end of the pipe before the device has gone offline or is powered off for other management purposes. The device that received the message shall disable the pipe immediately.

After the Device Pipe is disabled, the message based on previous pipe shall require setting up new pipe between the devices.

6.14 Device group dismiss and secession

When all IGRS Devices in the same peer device group quit from this device group, the peer-to-peer device group shall be dismissed (see 9.5.2).

The master device in a centralised device group may dismiss the device group according to management needs. When dismissing a centralised device group, the master device shall multicast a dismissing notification message to the network (see 9.5.3). After joining a centralised device group, an IGRS Device shall continuously monitor the online status of the device group. If one IGRS Device in a centralised device group receives the dismissing notification or detects the master device offline through the device online detection mechanism, it shall mean that the centralised device group has been dismissed. The device shall then quit from this device group.

An IGRS Device in a device group may initiate secession from the device group in the following cases.

- a) Device offline: after an IGRS Device joins the device group, if due to network disconnection or device power-off or for other device management reasons, it fails to multicast the device online advertisement to the network on a regular basis, it shall be regarded as device offline. The device offline status shall mean that the device quits from all device groups that it has joined.
- b) Initiated quit: after joining a centralised device group, an IGRS Device may send a Device Group secession request to the master device in the group through the Device Pipe in order to quit from the group. The quitting of an IGRS Device from a peer-to-peer device group is only an attribute change of the device itself, and does not require interaction with other peer devices in the group. An IGRS Device shall send a device online advertisement

message when leaving from the peer device group, and delete the related group's information from the online advertisement message (see 9.1.1).

6.15 Device offline

If IGRS Device A receives an offline advertisement message from IGRS Device B or does not receive an online advertisement message from device B again before the valid period of the last online advertisement of device B ends, A should regard B as offline.

The device online advertisement is based on the multicast mechanism. Due to the unreliable nature of a multicast mechanism, the device online status detection mechanism should be used to detect whether a device is online. After the Device Pipe between two IGRS Devices is established, either IGRS Device should send a device online detection message through the pipe to detect whether the IGRS Device on the other side is online. The IGRS Device that receives the message shall send an online detection response message back to the IGRS Device that initiates the detection. If the IGRS Device that initiates the detection fails to receive the response within a specified period, it should regard that device as offline.

If an IGRS Device in a centralised device group receives offline event notification about a particular device from the master device, that device should be regarded as offline.

7 IGRS message framework

7.1 Overview

The messages among IGRS Devices include three types: request messages, response messages, and notification messages.

The IGRS request/response messages shall conform to the request/response model of HTTP/1.1 (see RFC 2616), and also support all IGRS request/response messages through extended HTTP/1.1 command types and self-defined HTTP/1.1 headers (see 7.2, 7.3).

IGRS defines device notification messages that are uni-directional through extended HTTP/1.1 command types and self-defined HTTP/1.1 headers (see 7.2).

When authentication and encryption are needed, the framework of IGRS request/response and notification message differs from the framework without encryption (see 7.4).

When the transmission of IGRS request/response messages and notification messages do not adopt a chunked encoding transfer mechanism (RFC 2616), the message length (including the message body and message header) shall not exceed 20 kbyte. A message exceeding 20 kbyte shall adopt the chunked encoding transfer mechanism (RFC 2616) for transmission.

When an IGRS Device encounters an unknown field in a message, that field shall be ignored.

7.2 IGRS request/notification message structure

The IGRS request and notification message structure is shown in Table 1.

Table 1 – IGRS request and notification message

Message	Field explanation
POST[GET][NOTIFY] /IGRS HTTP/1.1	Extended HTTP command, including POST, NOTIFY and GET; the value of Request-URI part is /IGRS
Host: <i>Host IP Address:port</i>	HTTP header: required field
Content-type: text/xml: charset =utf-8	HTTP header: optional field
Content-Length: message body length	HTTP header, can be omitted when the message body is empty or transmitted by chunked encoding transfer mechanism
Man:" http://www.igrs.org/spec1.0 " ; ns= 01	IGRS extension header namespace definition: required field. See Clause B.1
01-IGRSVersion: 1.0	IGRS protocol version: required field
01-IGRSMessageType: <i>IGRS message type</i>	String: required field
01-SourceDeviceId: <i>ID of the device sends this message</i>	URI, defined in 8.1.2: required field
01-TargetDeviceId: <i>ID of the device receives this message</i>	URI, defined in 8.1.2: required field
01-SequenceId: <i>Sequence Number of this request message</i>	32 bit unsigned Int (0 is reserved): required field
Man:" http://www.w3.org/2002/12/soap-envelope "; ns=02	SOAP extension namespace definition: required field
02-SoapAction: <i>action identifier</i>	Refer to detailed message definition: required field
	Shall be blank line
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2002/12/soap-envelope" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">	
<SOAP-ENV:Body>	
<schema xmlns = "http://www.w3.org/2001/XMLSchema" xmlns:IGRS = "http://www.igrs.org/spec" targetNamespace = "http://www.igrs.org/spec1.0">	Name space definition, see Clauses B.9 and B.1
<element name="DeviceOperation" type="IGRS:DeviceOperationType minOccurs="0" maxOccurs="1"> <element name="Session" type="IGRS:SessionType"minOccurs="0" maxOccurs="1"> <complexType name="DeviceOperation"> <!--Device interaction--> </complexType> <complexType name="Session">	IGRS SOAP usage. ClientId, ServiceId and SequenceId are 32-bit unsignedInt (0 is reserved).

Message	Field explanation
<pre> <attribute name="request" type="IGRS:requestType" use="required"/> <!--Session interaction--> </complexType> <simpleType name="requestType"> <restriction base="xsd:string"> <xsd:enumeration value="NeedResponse"/> <xsd:enumeration value="NoResponse"/> </restriction> </simpleType> </schema> </pre>	
</SOAP-ENV:Body>	
</SOAP-ENV:Envelope>	
NOTE <i>Italics indicate where content is to be inserted; all other text in message definitions is fixed.</i>	

7.3 IGRS response message structure

The structure of IGRS response messages is shown in Table 2.

Table 2 – IGRS response message

Message	Field explanation
HTTP/1.1 200 OK	HTTP command line
Host: <i>Host IP Address:port</i>	HTTP message header
Content-type: text/xml; charset = utf-8	HTTP message header
Content-Length: message body length	HTTP header: can be omitted when the message body is empty or transmitted by chunked encoding transfer mechanism
Cache-Control: no-cache=ext;	HTTP message header
Ext:	HTTP message header
MAN:"http://www.igrs.org/spec1.0" ; ns= 01	IGRS namespace definition based on HTTP extension framework [IETF RFC 2774], see Clause B.1
01-IGRSVersion: 1.0	IGRS protocol version
01-IGRSMessageType: <i>IGRSMsgType</i>	string, refer to detailed message definition
01-SourceDeviceId: <i>ID of the device sends this message</i>	URI, defined in 8.1.2
01-TargetDeviceId: <i>ID of the device receives this message</i>	URI, defined in 8.1.2
01-ReturnCode: <i>Response status code</i>	Optional: defined in Clause 11
01-AcknowledgedId: <i>Acknowledged sequence number</i>	32 bit unsigned Int (0 is reserved), same with the SequenceId in request message
Man:"http://www.w3.org/2002/12/soap-envelope"; ns=02	SOAP HTTP extension identifier and header
02-SoapAction: <i>action identifier</i>	Refer to detailed message definition
	Shall be empty

Message	Field explanation
<code><SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2002/12/soap-envelope" SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"></code>	
<code><SOAP-ENV:Body></code>	
<code><schema xmlns = "http://www.w3.org/2001/XMLSchema" xmlns:IGRS = "http://www.igrs.org/spec" targetNamespace = "http://www.igrs.org/spec1.0"></code>	Name space, see Clauses B.9 and B.1
<code><element name="DeviceOperation" type="IGRS:DeviceOperationType minOccurs="0" maxOccurs="1"> <element name="Session" type="IGRS:SessionType"minOccurs="0" maxOccurs="1"> <complexType name="DeviceOperation"> <!--<i>Device interaction</i>--> </complexType> <complexType name="Session"> <!--<i>Session interaction</i>--> </complexType></code>	IGRS SOAP usage
<code></SOAP-ENV:Body></code>	
<code></SOAP-ENV:Envelope></code>	
NOTE Italics indicate where content is to be inserted; all other text in message definitions is fixed.	

7.4 IGRS message based on secure device pipe

7.4.1 Message authentication code generation

For devices implementing message interaction based on one of the following two mechanisms, the message authentication code shall be generated from the content of the IGRS header and the HTTP message body using the sequence and format defined in Table 3 and the corresponding algorithm in the adopted security mechanism, as follows:

- based on an identity authentication and message authentication mechanism;
- based on an identity authentication, encrypted message transmission, and message authentication mechanism of the Symmetric Key System (see 8.1.5.2 and 8.1.5.3).

The IGRS HTTP-extended header for labelling message authentication code is as follows:

01-MAC: hexadecimal ASCII character string to which the message authentication code corresponds.

Table 3 – Message authentication code

Message	Field explanation
01-IGRSVersion: IGRS/1.0	Not including CR for newline and blank
01-IGRSMessageType: <i>real message type</i>	Not including CR for newline and blank
01-SequenceId: <i>real sequence ID</i>	Not including CR for newline and blank; response message is 01-Acknowledged: real sequence ID; absent if there is no SequenceId/Acknowledged
01-SourceDeviceId: <i>real source device ID</i>	Not including CR for newline and blank
01-TargetDeviceId: <i>real target device ID</i>	Not including CR for newline and blank
Message body	HTTP message body content; message body format; absent if there is no message body
NOTE Italics indicate where content is to be inserted; all other text in message definitions is fixed.	

7.4.2 Message encryption

For devices implementing message interaction based on the identity authentication, encrypted message transmission, and message authentication mechanism of the Symmetrical Key System (see 8.1.5.2, 8.1.5.3), the encryption object is the original HTTP message body, which is based on secure Device Pipe interaction.

The content-length and content-type of the HTTP message header based on secure Device Pipe interaction shall follow the following rules.

- Content-type: application/octet-stream.
- Content-length: content length of message body after encryption.
- Encryption shall not be needed if there is no message body.

8 IGRS device and service description

8.1 IGRS device description

8.1.1 Device description template

An IGRS Device describes relevant information such as device name, type, and service list with one XML document that conforms to the IGRS Device Description Template specified below.

An IGRS Client may get the device description documents of other IGRS Devices through the request mechanism specified in 9.3.

The IGRS Device description template is defined as follows (see also Clause B.5):

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<schema targetNamespace="http://www.igrs.org/igrs/DeviceTemplate"
```

```
xmlns:igrs="http://www.igrs.org/igrs/DeviceTemplate"
```

```
xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
```

```
attributeFormDefault="unqualified">
```

```
<element name="root">
```

```

<complexType>
    <sequence>
<element name="deviceTemplateVersion" type="igrs:deviceTemplateVersionType"/>
        <element name="device" type="igrs:deviceTemplateType"/>
            <element name="protocolVersion"
type="igrs:protocolVersionType"/>
        </sequence>
    </complexType>
</element>

<complexType name="deviceTemplateVersionType">
    <sequence>
        <element name="major" type="unsignedInt"/>
        <element name="minor" type="unsignedInt"/>
    </sequence>
</complexType>

<complexType name="protocolVersionType">
    <sequence>
        <element name="major" type="unsignedInt"/>
        <element name="minor" type="unsignedInt"/>
    </sequence>
</complexType>

<complexType name="deviceTemplateType">
    <sequence>
        <element name="deviceType" type="anyURI"/>
        <element name="deviceName" type="string"/>
        <element name="manufacturer" type="string"/>
        <element name="manufacturerURL" type="anyURI" minOccurs="0"/>
        <element name="modelDescription" type="string" minOccurs="0"/>
    </sequence>
</complexType>

```

```
<element name="modelName" type="string"/>

<element name="modelName" type="string" minOccurs="0"/>

<element name="modelURL" type="anyURI" minOccurs="0"/>

<element name="serialNumber" type="string" minOccurs="0"/>

<element name="UDN" type="anyURI"/>

<element name="DeviceIconURL" type="anyURI"/>

<element name="joinedGroupList" type="igrs:joinedGroupListType"
minOccurs="0"/>

<element name="deviceSecurityIdList"
type="igrs:deviceSecurityIdListType"/>

    <element name="serviceList"
type="igrs:serviceListType"minOccurs="0"/>

</sequence>

</complexType>

<complexType name="joinedGroupListType">

    <sequence>

        <element name="deviceGroup" type="igrs:deviceGroupType"/>

    </sequence>

</complexType>

<complexType name="deviceGroupType">

    <sequence>

        <element name="deviceGroupId" type="anyURI"/>

        <element name="deviceGroupName" type="string"/>

    </sequence>

</complexType>

<complexType name="deviceSecurityIdListType">

    <sequence>
```

```

        <element name="deviceSecurityId" type="anyURI" minOccurs="0"
maxOccurs="5"/>

    </sequence>

</complexType>

<complexType name="serviceListType">

    <sequence>

        <element name="service" type="igrs:serviceTemplateType"
maxOccurs="unbounded"/>

    </sequence>

</complexType>

<complexType name="serviceTemplateType">

    <sequence>

        <element name="serviceType" type="anyURI"/>

        <element name="serviceId" type="unsignedInt"/>

        <element name="serviceName" type="string"/>

        <element name="serviceSecurityIdList"

type="igrs:serviceSecurityIdListType"/>

        <!-- Application developer may add additional service interfaces, but should use different
namespaces to make a distinction -->

    </sequence>

</complexType>

<complexType name="serviceSecurityIdListType">

    <sequence>

        <element name="serviceSecurityId" type="anyURI" minOccurs="0"
maxOccurs="3"/>

    </sequence>

</complexType>

</schema>

```

8.1.2 Device identifier

An IGRS Device identifier is a globally unique ID, called a UUID, so no two devices have the same ID.

The format of an IGRS Device identifier is described as follows by adopting the ABNF (IETF RFC 2234):

`<IGRSDeviceIdURN> ::= "urn:" <IGRSNS> "." <IGRSIDVAL>`

`<IGRSNS> ::= "IGRS:Device:DeviceId"`

`<IGRSIDVAL> ::= 8 <URN chars> "-" 4<URN chars> "-" 4<URN chars> "-" 4<URN chars> "-" 12<URN chars>`

`<URN chars> ::= <trans>`

`<trans> ::= <upper> | <lower> | <number>`

`<upper> ::= "A" | "B" | "C" | "D" | "E" | "F"`

`<lower> ::= "a" | "b" | "c" | "d" | "e" | "f"`

`<number> ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" |
"8" | "9"`

`<URN chars>` is case-insensitive.

8.1.3 Device group identifier

An IGRS Device group identifier is a globally unique ID so no two groups have the same ID.

The format of an IGRS Device group identifier is described as follows by adopting the ABNF (RFC 2234):

`<IGRSDeviceGroupIdURN> ::= "urn:" <IGRSNS> "." <IGRSIDVAL>`

`<IGRSNS> ::= "IGRS:DeviceGroup:DeviceGroupId"`

`<IGRSIDVAL> ::= 8 <URN chars> "-" 4<URN chars> "-" 4<URN chars> "-" 4<URN chars> "-" 12<URN chars>`

`<URN chars> ::= <trans>`

`<trans> ::= <upper> | <lower> | <number>`

`<upper> ::= "A" | "B" | "C" | "D" | "E" | "F"`

`<lower> ::= "a" | "b" | "c" | "d" | "e" | "f"`

`<number> ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" |
"8" | "9"`

<URN chars> is case-insensitive.

8.1.4 Device type identifier

An IGRS Device type identifier is used for identifying an IGRS Device set containing the same functions. The length of the IGRS Device type identifier shall be no more than 127 bytes.

The format of the IGRS Device type identifier is shown as follows by adopting the ABNF (RFC 2234):

```
<IGRSDeviceTypeURN>::="urn:"<IGRSNS>":"<IGRSSingleType>|<IGRSTypeList>
```

```
<IGRSNS>::="IGRS:Device:DeviceType"
```

```
<IGRSTypeList>=<IGRSSingleType>*<IGRSTypeVal>
```

```
<IGRSTypeVal>=<ConnectionSign><IGRSSingleType>
```

```
<IGRSSingleType>= <NAME>
```

```
<NAME>::=1*16<URN chars>
```

```
<URN chars>::=<trans>
```

```
<trans>::=<upper>|<lower>|<number>|<other>
```

```
<upper>::="A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" |
```

```
    "I" | "J" | "K" | "L" | "M" | "N" | "O" | "P" |
```

```
    "Q" | "R" | "S" | "T" | "U" | "V" | "W" | "X" |
```

```
    "Y" | "Z"
```

```
<lower>::="a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" |
```

```
    "i" | "j" | "k" | "l" | "m" | "n" | "o" | "p" |
```

```
    "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" |
```

```
    "y" | "z"
```

```
<number>::="0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" |
```

```
    "8" | "9"
```

```
<other>::="-" | "." | "_"
```

```
<ConnectionSign>::=";"
```

<URN chars> is case-insensitive.

8.1.5 Device security mechanism descriptor

8.1.5.1 Null security mechanism among devices

When two IGRS Devices that set up the Device Pipe adopt this security mechanism, no security assurance measures shall be taken. That is, when the pipe is established, ID authentication for the device shall not be performed. Messages in the channel are transmitted without using IGRS specifications for confidentiality and/or integrity. This mechanism shall be the required default security mechanism for IGRS Devices and may be used in an application scenario where data security is not needed, for example, when communicating on a wired medium within a house.

IGRS based on this device security mechanism is described as follows by adopting the ABNF (RFC 2234):

<IGRSDeviceSecurityURN> :: ="urn:"<IGRSNS>

<IGRSNS> :: ="IGRS:DeviceSecurity:NULL"

8.1.5.2 Identity authentication and message authentication mechanism based on symmetric-key cryptosystem

Two IGRS Devices may use a pre-shared key to authenticate each other and to establish a secure Device Pipe. The authentication process is a bi-directional challenge/response process. The message transmitted on the Device Pipe shall not be encrypted, but shall be checked for integrity and authenticity. This mechanism is optional for IGRS Devices and may be applied in small networks and devices with weak cipher computing capabilities. This security option might be useful in a scenario where bogus or unqualified devices shall be excluded from a device group. This security method protects the devices that are communicating, not the data transmitted.

The challenge value in the process of the ID authentication in this mechanism shall be generated based on the secure random number generator (RNG) algorithm or pseudo-random number generator (PRNG) algorithm and shall be regenerated each time when performing ID authentication. The design and implementation of the RNG or PRNG algorithm conforming to the general detection standard is outside the scope of this standard. Please see ISO/IEC 18031:2005 for the required elements and characteristics of RNG and PRNG.

The response value in the process of ID authentication in this mechanism is generated through the secure hash algorithm based on the other party's challenge value and the preset shared key. The generation method is defined as follows.

- The secure hash algorithm function is Hash (x).
- The response value is Hash (Challenge || Key), where Challenge is the challenge value and Key is the shared key by both parties.

The secure Hash algorithms supported by this mechanism are SHA-1 and SHA-256 specified in ISO/IEC 10118-3:2004.

The mechanism used for ensuring the integrity and authentication of the message that is transmitted in the pipe is the following.

- a) Calculate one digest value for the target message stream transmitted by the pipe using the secure Hash function.
- b) Calculate the digest value again after concatenating with the preset-shared key.
- c) Attach the digest value to 01-MAC.
- d) Send the digest value and 01-MAC to the receiver.

This digest value shall be used as the basis for determining the integrity and authentication of the message. The algorithm format is: Hash (Hash (message) || Key), where, the Message is the message to be transmitted by the pipe and the Key is the shared key for both parties.

After receiving the message, the receiver shall perform the same calculation process, and shall compare the calculation results with the added digest value. If they are equal, the message is transmitted by the transmitter without modification; otherwise, the message shall be regarded as untrusted.

The precondition for implementing this mechanism is that both protocol participants should share one key and establish an appropriate system outside of IGRS to complete the secure distribution and management of the shared key.

IGRS based on this device security mechanism, is described as follows by adopting the ABNF (RFC 2234):

<IGRSDeviceSecurityURN>::="urn:" <IGRSNS> ":" <IGRSVAL1> ":" <IGRSVAL2>

<IGRSNS>::="IGRS:DeviceSecurity:PreSharedKey_MAC"

<IGRSVAL1>::="SHA-1"|"SHA-256"

<IGRSVAL2>::="SHA-1"|"SHA-256"

IGRSVAL1 is the description of the identity authentication algorithm; IGRSVAL2 is the description of the message authentication algorithm.

8.1.5.3 Identity authentication, encrypted message transmission and message authentication mechanism based on symmetric-key cryptosystem

When two IGRS Devices adopt this security mechanism to establish the Device Pipe, ID authentication shall be performed for the two devices based on a pre-shared key. The authentication process is a bi-directional challenge/response process. This security mechanism provides increased confidentiality, integrity, and authenticity for messages transmitted on the device channel. This mechanism is optional for IGRS Devices and may be applied in small networks and devices with weak cipher computing capabilities. This security option might be useful in a scenario where bogus or unqualified devices must be excluded from a device group. This security method may be useful for a wireless medium because it protects the devices that are communicating and the data transmitted,

The challenge value, Key, or Nonce in this mechanism shall be generated using the secure RNG or PRNG algorithm. The secure hash algorithm shall be used to implement the challenge/response authentication protocol and message integrity/authenticity protocol. Meanwhile, the method of ensuring the integrity and authenticity of the messages transmitted in the pipe is also the same as the second security mechanism (specified in 8.1.5.2).

In addition, this security mechanism uses the symmetric-key cipher algorithm to ensure the confidentiality of messages. The PipeKey that implements the message confidentiality is generated through a Hash algorithm in the process of protocol authentication based on the two challenge values generated by the bi-directional challenge/response protocol and the pre-shared keys. The generation method is:

- a) if the length of the bit string output by the Hash (Hash (Challenge 1|| Challenge 2) || Key) is greater than or equal to the length of key of the symmetric-key cipher algorithm, the bit string with the same length as the key length drawn from left to right shall be used as the PipeKey;
- b) if the length of the bit string output by the Hash (Hash (Challenge 1|| Challenge 2) || Key) is less than the length of key of the symmetric-key cipher algorithm,

(Hash(Hash(Challenge1 || Challenge 2)|| Key) || Hash (Hash (Hash (Challenge 1 || Challenge 2) || Key)|| Key)) is generated;

- c) if the length is greater than or equal to the length of key of the symmetric-key cipher algorithm, the bit strings with the same length as the key length drawn from left to right shall be used as the PipeKey.

The rest shall follow the same rules. The PipeKey, which can be shared by both parties, shall then be generated.

The PipeKey is only valid during the life cycle of the current pipe.

When implementing a symmetric-key cipher block algorithm (see ISO/IEC 18033-3), CBC (Cipher Block Chaining) mode for encryption/decryption is used. When the message is divided into blocks, the length of the last block may be insufficient and will need padding. The padding method is the following: assume the encrypted message length is l (unit is byte) and the block size is n bytes. 1 byte will be needed to represent the length of the padding (unit is byte) and u bytes for padding (every bit is 0)., u is the least non-negative integer ($0 \leq u < 16$) that satisfies $(l + 1 + u) \bmod n = 0$. The last byte in the block will be used to represent the padding length ($u + 1$). When using the CBC mode to encrypt/decrypt messages, the first message block will perform an XOR operation with an Initialisation Vector (IV), then apply block encryption algorithm to encrypt the message. The result will become the second "IV." This operation will continue for the remaining blocks. Here, IV is randomly selected by the encryption side before the encryption algorithm begins. The length of the IV is the block length used in the block encryption algorithm. IV will be placed in front of the encryption result of the first block to send to the receiving end. The mathematical formulas used by CBC encryption/decryption mode are: encryption: $C_1 = E_K(P_1 \oplus IV)$, $C_2 = E_K(P_2 \oplus C_1)$, ..., $C_N = E_K(P_N \oplus C_{N-1})$; decryption: $P_1 = IV \oplus D_K(C_1)$, $P_2 = C_1 \oplus D_K(C_2)$, ..., $P_N = C_{N-1} \oplus D_K(C_N)$. Here N means the number of blocks in the message.

The symmetric-key cipher algorithms that this security mechanism currently supports are

- AES-128-128 (128-bit block-length and 128-bit key-length), specified in ISO/IEC 18033-3,
- 3DES-64-112 (64-bit block-length and 112-bit key-length), specified in ISO/IEC 19790.

The implementation of this security mechanism to enable the secure distribution and management of a shared key is outside the scope of this standard.

IGRS based on this device security mechanism, is described as follows by adopting the ABNF (RFC 2234):

<IGRSDeviceSecurityURN>::="urn:"<IGRSNS>":"<IGRSVAL1>":"<IGRSVAL2>":"

<IGRSVAL3>

<IGRSNS>::="IGRS:DeviceSecurity:PreSharedKey_Cipher_MAC"

<IGRSVAL1>::="SHA-1"|"SHA-256"

<IGRSVAL2>::="AES-128-128"|"3DES-64-112"

<IGRSVAL3>::="SHA-1"|"SHA-256"

IGRSVAL1 is the description of the identity authentication algorithm; IGRSVAL2 is the description of the message encryption algorithm; IGRSVAL3 is the description of the message authentication algorithm.

8.1.5.4 Identity authentication, encrypted message transmission and message authentication mechanism based on public-key cryptosystem

When two IGRS Devices adopt this security mechanism to establish a Device Pipe, the two devices shall be authenticated based on the PKI X.509 v3 (ISO/IEC 9594-8:2005). The authentication process is a bi-directional challenge-response process. This security mechanism provides increased confidentiality, integrity, and authenticity for messages transmitted on the device channel. This mechanism is optional for IGRS Devices and may be applied in medium or large networks and devices with strong cipher computing capabilities. This security option might be useful in a scenario where bogus or unqualified devices shall be excluded from a device group. This security method may be useful for a wireless medium because it protects the devices that are communicating and the data transmitted.

NOTE A public key is an asymmetric key system where one key is published and the other is secret. This could support mass-marketed devices all containing the same public key, while the secret key is embedded in or downloaded into a controller communicating with the mass-marketed devices.

The challenge value, Key, or Nonce in this mechanism shall be generated using the secure RNG or PRNG algorithm. The secure hash algorithm shall be used to enable the message integrity/authenticity protocol.

For integrity and authenticity when processing a message, the secure hash algorithm shall be used to calculate the message to produce a digest. Then a private key shall be used corresponding to the PKI X.509 certificate to perform signature operation on the digest value. The signature result shall be the criterion for message integrity and authenticity. The signature results together with the message shall be sent to the message receiver.

The PipeKey generating method for message confidentiality in this mechanism is different from the third security mechanism. The following two design methods may be considered.

- a) Simple method: The Pipe Key is generated from negotiations during the bi-directional challenge/response authentication process. The detailed procedure is the following.
 - The response party in the first round selects one random number (reserved for the PipeKey).
 - Encrypts it with the other party's public key.
 - Sends the encryption results, response value, and the challenge value for the second round together to the other party.
 - The response party in the second round uses its private key to decrypt the encrypted value and to perform a secure Hash algorithm for the concatenation of the decryption result and the challenge value. Meanwhile, the response party uses the private key to generate a signature for the challenge value and sends the signature result together with the result of the Hash algorithm to the other party.
 - The receiver party determines that the other party has obtained accurate (used as the PipeKey) random number through verifying the signature and the comparison with the result of the Hash algorithm.
 - The random number shall be used as the shared key PipeKey for both parties to enable message confidentiality.
- b) DH method: Adopt the Diffie-Hellman key switching protocol to negotiate the key PipeKey shared by both parties. In this case, the Diffie-Hellman public key cipher model and relevant algorithm shall be constructed.

With the third security mechanism, the PipeKey is only valid during the current life cycle of the pipe and the PipeKey updating mechanism is not considered either.

The symmetric-key cipher algorithms, which this security mechanism currently supports, are AES and 3DES. The public key algorithms, which this security mechanism currently supports, are RSA-1024 [IETF RFC 3447] and ECC-192 (IEEE 1363-2000).

The enabling of this security mechanism needs the support of the PKI X.509 v3 certificate and CA (Certificate Authority) system. Due to the large scalability of the system, when the IGRS protocol network is enabled within a small scope, the developers of protocols can enable one simplified and sufficient X.509 CA system. When the IGRS protocol network is implemented in a large-scale network, a complicated X.509 CA system is needed, or an external existing X.509 CA system is another option.

It is required that the IGRS protocol use the public key certificate function of this security mechanism according to the X.509 CA system interface standard.

IGRS based on this device security mechanism is described as follows by adopting the ABNF (RFC 2234):

```
<IGRSDeviceSecurityURN>::="urn:"<IGRSNS>":"<IGRSVAL1>":"<IGRSVAL2>":"
```

```
<IGRSVAL3>":"<IGRSVAL4>
```

```
<IGRSNS>::="IGRS:DeviceSecurity:PKICertificate_Cipher_MAC"
```

```
<IGRSVAL1>::="RSA-1024"|"ECC-192"|"ElGamal"
```

```
<IGRSVAL2>::="Simple"|"DH"
```

```
<IGRSVAL3>::="AES-128-128"|"3DES-64-112"
```

```
<IGRSVAL4>::="SHA-1"|"SHA-256"
```

IGRSVAL1 is the description of the public key algorithm; IGRSVAL2 is the description of the negotiation key algorithm; IGRSVAL3 is the description of encrypted message transmission algorithm, and IGRSVAL4 is the description of message authentication algorithm.

8.1.5.5 Identity authentication, encrypted message transmission and message signature mechanism based on trusted third party

This security mechanism provides the ID authentication service for both parties that set up the pipe based on the trusted third party. The three parties complete the authentication process according to the Kerberos v5 protocol [IETF RFC 1510]. The key generation and distribution function for message confidentiality, integrity and authenticity shall be provided in the protocol ticket generated by the trusted third party. Generally, this mechanism should allow handling with the confidentiality, integrity, and authenticity of messages transmitted in the pipe, but also can support the two parties of a pipe to negotiate and select according to the security requirement stipulated by the device advertisement. This security mechanism is optional for IGRS Devices and may be applied in large enterprise networks and systems dedicated to support security. This security option might be useful in a scenario where bogus or unqualified devices must be excluded from a device group. This security method may be useful for a wireless medium because it protects the devices that are communicating and the data transmitted.

NOTE The involvement of a trusted third party to supply keys is commonly used by web browsers in Internet commerce to block secure communications with bogus web sites masquerading as known vendors. A trusted key is important if home control is offered as a managed service from a web-based vendor.

There are two methods for implementing the Kerberos protocol specified in IETF RFC 1510:

One is based on the shared key or symmetric key encryption mechanism. In this case, within the scope of the IGRS protocol network, each IGRS Device simply shares one key with the trusted third party (authentication service provider) in advance, without pre-setting the shared key between the two interworking devices.

The other implementation method is based on the public key system or PKI X.509 CA certificate system. Although this method could be designed as the fourth security mechanism, computation limitations should be considered. Since some low-power computing platforms may have difficulty implementing the public key encryption algorithm, the mixed mode is generally considered. In the mixed mode the platform with the more powerful computing capability adopts the PKI X.509 certificate system, while the platform with the less powerful computing capability adopts the symmetric-key cipher system. The trusted third party that offers the authentication services should be able to support both cipher systems. The security mechanism between the device and the trusted third party should adopt the IGRS Device security mechanism defined in this document.

The trusted third party-based authentication services and the Kerberos protocol are highly efficient only when an IGRS network has a large scale (especially the certificate system based on PKI X.509). The concept of the centralised device group also exists in the IGRS protocol network, and the master device may assume the role of the trusted third party. Therefore, for this security mechanism, the enabling method using the symmetric-key cipher system is recommended.

From the perspective of the development trend in security technology, authentication services based on the trusted third party are becoming main stream. Meanwhile, the technical complexity is great. Therefore, this security mechanism shall be considered as the basis or direction for future extensions of the security technology of the IGRS protocol. At present, it is recommended that a simplified and effective method be chosen to enable this security mechanism, i.e., only support the minimum set of the Kerberos v5 mechanism attributes.

This security mechanism also needs the support of the secure RNG or PRNG algorithm, secure hash algorithm, and symmetric-key cipher algorithm.

The symmetric-key cipher algorithms, which this security mechanism currently supports, are AES and 3DES. The public key algorithms, which this security mechanism currently supports, is RSA-1024.

IGRS based on this device security mechanism, is described as follows by adopting the ABNF (RFC 2234):

<IGRSDeviceSecurityURN>::="urn:"<IGRSNS>":"<IGRSVAL1>":"<IGRSVAL2>":"

<IGRSVAL3>":"<IGRSVAL4>":"<IGRSVAL5>":"<IGRSVAL6>

<IGRSNS>::="IGRS:DeviceSecurity:3rdPartyAuthenService"

<IGRSVAL1>::="3rdAuthProtocol_Kerberosv5:AES-128-128"|3rdAuthProtocol_Kerberosv5:RSA-1024"

<IGRSVAL2>::="AuthenServiceProvider"

<IGRSVAL3>::=<IGRSDeviceIdURN> (see 8.1.2)

<IGRSVAL4>::= Authentication service identifier (see 8.2.2)

<IGRSVAL5>::="AES-128-128"|3DES-64-112"

<IGRSVAL6>::="SHA-1"|SHA-256"

IGRSVAL1 is the description of the Kerberos5-based authentication algorithm of the trusted third party mechanism. IGRSVAL2, 3 and 4 compose the description of the third party authentication service provider, where IGRSVAL3 is the identifier for device that provides

authentication service and IGRSVAL4 is the service identifier of such service on a device that is identified by IGRSVAL3. IGRSVAL5 is the description of the encrypted message transmission algorithm for both end devices of a secure pipe, and IGRSVAL6 is the description of message signature algorithm for both end devices of a secure pipe.

8.1.5.6 Encryption algorithm descriptors supported by security mechanism

The five security mechanisms described in 8.1.5.1 through 8.1.5.5 are categorized below by the following levels of complexity from the lowest (level 0) to the highest complexity (level 4). Level 0 is the default and means no security is used.

0. NULL

1. PreSharedKey_MAC

2. PreSharedKey_Cipher_MAC

3. PKICertificate_Cipher_MAC

4. 3rdPartyAuthenService

Every device shall implement the null security mechanism. The other four mechanisms are optional. However, if a device implements a security level above 0, it shall implement all lower-complexity levels. For an example, a device that implements PKICertificate_Cipher_MAC must also implement PreSharedKey_Cipher_MAC, PreSharedKey_MAC, and NULL mechanisms. When a communications session via a device pipe is initiated, the following procedure shall be executed to ensure that the same and most secure mechanism is used by the devices sharing the pipe:

- a) Using the NULL level of security, each device shall announce to the other devices or to the master, if present, the highest level of security it can support;
- b) the device that initiates the secure pipe setup or the master shall determine the highest level supported by the devices sharing the pipe based on the security algorithm parameters it receives from the other devices;
- c) the initiating device or the master shall send to each device a secure pipe setup request message to use this level for subsequent communications via the pipe;
- d) a secure pipe setup response message shall be returned from the recipient device to the initiating device or to the master verifying the successful device pipe setup at the specified security level.

Table 4 describes five security mechanisms defined by IGRS and the corresponding relationships between the security protocol and the security algorithm used in each mechanism.

Table 4 – Device security mechanism protocol algorithm

Device security mechanism	Security algorithm type involved	Algorithm related to device security mechanism
PreSharedKey_MAC	AuthenticationAlgorithm	AuthenticationAlgorithm:SHA-1
		AuthenticationAlgorithm:SHA-256
	MACAlgorithm	MACAlgorithm:SHA-1
		MACAlgorithm:SHA-256
PreSharedKey_Cipher_MAC	AuthenticationAlgorithm	AuthenticationAlgorithm:SHA-1
		AuthenticationAlgorithm:SHA-256
	EncryptAlgorithm	EncryptAlgorithm:AES-128-128
		EncryptAlgorithm:3DES-64-112
	MACAlgorithm	MACAlgorithm:SHA-1
		MACAlgorithm:SHA-256
PKICertificate_Cipher_MAC	PublicKeyAlgorithm	PublicKeyAlgorithm:RSA-1024
		PublicKeyAlgorithm:ECC-192
		PublicKeyAlgorithm:ElGamal
	KeyNegotiationAlgorithm	KeyNegotiationAlgorithm:Simple
	EncryptAlgorithm	EncryptAlgorithm:AES-128-128
		EncryptAlgorithm:3DES-64-112
	MACAlgorithm	MACAlgorithm:SHA-1
		MACAlgorithm:SHA-256
3 rd PartyAuthenService	3 rd AuthProtocol_Kerberosv5	3 rd AuthProtocol_Kerberosv5:AES-128-128
		3 rd AuthProtocol_Kerberosv5:RSA-1024
	AuthenServiceProvider	AuthenServiceProvider: DeviceID ServiceID
	EncryptAlgorithm	EncryptAlgorithm:AES-128-128
		EncryptAlgorithm:3DES-64-112
	MACAlgorithm	MACAlgorithm:SHA-1
		MACAlgorithm:SHA-256

The level of complexity design principle for implementing security mechanisms shall also apply to each of the security algorithm options in the third column of Table 4. These options are listed in order from lowest to highest complexity. All lower complexity security algorithms must be implemented whenever a higher complexity security algorithm is chosen. For example, if the AuthenticationAlgorithm of the PreSharedKey_MAC is required, then a device that implements SHA-256 shall also implement SHA-1.

8.2 IGRS service description

8.2.1 Service description template

The IGRS Service description is based on WSDL2.0 (specified in <http://www.w3.org/TR/wsdl20/>). In addition, the IGRS Service description adds specific requirements that include:

- extension of WSDL portType, add support to IGRS Service information description;
- transmission Binding based on Device Pipe.

This standard (see Clause 10) extends the WSDL service description template specified in W3C WSDL 2.0. The IGRS Service description template is shown as follows (see also Clause B.3):

```

<wsdl:definitions xmlns:igrs="http://www.igrs.org/igrs/ServiceDescription"

xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://www.igrs.org/igrs/ServiceDescription" name="igrs">

    <wsdl:types>

<schema targetNamespace=http://www.igrs.org/igrs/ServiceDescription

xmlns="http://www.w3.org/2001/XMLSchema"

xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"

attributeFormDefault="unqualified" elementFormDefault="qualified">

        <!-- IGRS Service Attribute Types -->

        <attributeGroup name="occurs">

            <attribute name="minOccurs" type="xsd:nonNegativeInteger"

use="optional" default="1"/>

            <attribute name="maxOccurs">

                <simpleType>

                    <union

memberTypes="xsd:nonNegativeInteger">

                        <simpleType>

                            <restriction base="NMTOKEN">

                                <enumeration

value="unbounded"/>

                                    </restriction>

                                </simpleType>

                            </union>

                        </simpleType>

                    </attribute>

                </attributeGroup>

```

```

<element name="serviceAttribute" type="igrs:IGRSDDataType"/>

<element name="serviceAttributeValues"
type="igrs:IGRSDDataValueType"/>

<complexType name="IGRSDDataType">

    <sequence>

        <element name="ServiceId" type="xsd:unsignedInt"/>

        <element name="ServiceName" type="xsd:string"/>

        <element name="ServiceType" type="igrs:ServiceType"/>

        <element
name="ServiceSecurityId" type="igrs:ServiceSecurityId"/>

        <!--other elements for serviceAttribute -->

    </sequence>

    <attribute name="name" type="xsd:NCName"/>

    <attribute name="type" type="xsd:QName"/>

    <attributeGroup ref="igrs:occurs"/>

    <attribute name="mutability" use="optional" default="dynamic">

        <simpleType>

            <restriction base="xsd:string">

                <enumeration value="static"/>

                <enumeration value="dynamic"/>

            </restriction>

        </simpleType>

    </attribute>

    <attribute name="notifiable" use="optional" type="xsd:boolean"
default="false"/>

</complexType>

<complexType name="IGRSDDataValueType">

    <sequence>

```

```
maxOccurs="unbounded"/>
    <any namespace="##any" minOccurs="0"
    </sequence>
  </complexType>
  <!-- findServiceAttribute Message Types -->
  <element name="findServiceAttributeByNames"
type="igrs:NCNamesType"/>
  <element name="subscribeServiceAttributeByNames"
type="igrs:NCNamesType"/>
  <element name="findServiceAttribute"
type="igrs:findServiceAttributeType"/>
  <complexType name="findServiceAttributeType">
    <sequence>
      <element name="findExpression"
type="igrs:findExpressionType"/>
    </sequence>
  </complexType>
  <complexType name="findExpressionType">
    <sequence>
      <element ref="igrs:findServiceAttributeByNames"/>
    </sequence>
  </complexType>
  <complexType name="NCNamesType">
    <sequence>
      <element name="name" type="xsd:NCName"
maxOccurs="unbounded"/>
    </sequence>
  </complexType>
  <element name="findServiceAttributeResponse"
```

type="igrs:findServiceAttributeResponseType"/>

<complexType name="findServiceAttributeResponseType">

<sequence>

<element name="result" type="igrs:resultType"/>

</sequence>

</complexType>

<complexType name="resultType">

<sequence>

<element name="serviceAttributeValues"

type="igrs:serviceAttributeValuesType" minOccurs="0"/>

<element name="returnCode" type="xsd:unsignedInt"/>

</sequence>

</complexType>

<complexType name="serviceAttributeValuesType">

<sequence maxOccurs="unbounded">

<element name="name" type="xsd:NCName"/>

<element name="value"

type="igrs:IGRSDataValueType"/>

</sequence>

</complexType>

<!-- Subscription Message Types -->

<element name="subscribe" type="igrs:subscribeType"/>

<complexType name="subscribeType">

<sequence>

<element name="subscribeExpression"

type="igrs:subscribeExpressionType"/>

<element name="location" type="xsd:anyURI"/>

<element name="expirationTime" type="xsd:dateTime"/>

```

        </sequence>

    </complexType>

    <complexType name="subscribeExpressionType">

        <sequence>

            <element
ref="igrs:subscribeServiceAttributeByNames"/>

        </sequence>

    </complexType>

    <element name="renewSubscription">

        <complexType>

            <sequence>

                <element name="subscriptionId"
type="xsd:unsignedInt"/>

                <element name="location" type="xsd:anyURI"/>

                <element name="expirationTime"
type="xsd:dateTime"/>

            </sequence>

        </complexType>

    </element>

    <element name="unsubscribe">

        <complexType>

            <sequence>

                <element name="subscriptionId"
type="xsd:unsignedInt"/>

            </sequence>

        </complexType>

    </element>

    <element name="subscribeResponse"
type="igrs:subscribeResponseType"/>

    <complexType name="subscribeResponseType">
```

```

<sequence>
    <element name="subscriptionId"
type="xsd:unsignedInt"/>
    <element name="terminationTime" type="xsd:dateTime"
minOccurs="0"/>
    <element name="returnCode" type="xsd:unsignedInt"/>
</sequence>
</complexType>
<!-- Notification Message Types -->
<element name="sendNotification" type="igrs:sendNotificationType"/>
<complexType name="sendNotificationType">
    <sequence>
        <element name="serviceAttribute"
type="igrs:serviceAttributeTypes"/>
    </sequence>
</complexType>
<complexType name="serviceAttributeTypes">
    <sequence maxOccurs="unbounded">
        <element name="name" type="xsd:NCName"/>
        <element name="value"
type="igrs:IGRSDataValueType"/>
        <element name="subscriptionId"
type="xsd:unsignedInt"/>
    </sequence>
</complexType>
<import namespace="http://schemas.xmlsoap.org/wsdl"/>
<element name="IGRSportType" type="igrs:portTypeType"/>
<complexType name="portTypeType">
    <complexContent>

```

```
<extension base="wsdl:tPortType">

    <sequence>

        <any namespace="##other"
minOccurs="0"
maxOccurs="unbounded"/>

    </sequence>

    <attribute name="extends" use="optional">

        <simpleType>

            <list itemType="QName"/>

        </simpleType>

    </attribute>

    <anyAttribute namespace="##other"/>

</extension>

</complexContent>

</complexType>

</schema>

</wsdl:types>

<message name="findServiceAttributeInputMessage">

    <part name="parameters" element="igrs:findServiceAttribute"/>

</message>

<message name="findServiceAttributeOutputMessage">

    <part name="parameters" element="igrs:findServiceAttributeResponse"/>

</message>

<message name="subscribeInputMessage">

    <part name="parameters" element="igrs:subscribe"/>

</message>

<message name="subscribeOutputMessage">

    <part name="parameters" element="igrs:subscribeResponse"/>
```

```

</message>

<message name="sendNotificationInputMessage">
    <part name="parameters" element="igrs:sendNotification"/>
</message>

<IGRSPORTType name="subscribeServiceAttributePortType">
    <operation name="subscribe">
        <input message="igrs:subscribeInputMessage"/>
        <output message="igrs:subscribeOutputMessage"/>
    </operation>
    <igrs:serviceAttribute name="notifiableServiceAttributeName"
type="igrs:IGRSDDataType"
minOccurs="0" maxOccurs="unbounded" mutability="dynamic" notifiable="true"/>
    <igrs:staticServiceAttributeValues>
        <igrs:subscribeExtensibility
inputElement="igrs:subscribeServiceAttributeByNames"/>
    </igrs:staticServiceAttributeValues>
</IGRSPORTType>

<IGRSPORTType name="notifyServiceAttributePortType">
    <operation name="sendNotification">
        <input message="igrs:sendNotificationInputMessage"/>
    </operation>
</IGRSPORTType>

<IGRSPORTType name="findServiceAttributePortType">
    <operation name="findServiceAttribute">
        <input message="igrs:findServiceAttributeInputMessage"/>
        <output message="igrs:findServiceAttributeOutputMessage"/>
    </operation>

```