ONVIF™
Authentication Behavior Specification

Version 19.12
December 2019

THIS DOCUMENT IS PROVIDED "AS IS," AND THE CORPORATION AND ITS MEMBERS AND THEIR AFFILIATES, MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THIS DOCUMENT ARE SUITABLE FOR ANY PURPOSE; OR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

IN NO EVENT WILL THE CORPORATION OR ITS MEMBERS OR THEIR AFFILIATES BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES, ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THIS DOCUMENT, WHETHER OR NOT (1) THE CORPORATION, MEMBERS OR THEIR AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR (2) SUCH DAMAGES WERE REASONABLY FORESEEABLE, AND ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THIS DOCUMENT. THE FOREGOING DISCLAIMER AND LIMITATION ON LIABILITY DO NOT APPLY TO, INVALIDATE, OR LIMIT REPRESENTATIONS AND WARRANTIES MADE BY THE MEMBERS AND THEIR RESPECTIVE AFFILIATES TO THE CORPORATION AND OTHER MEMBERS IN CERTAIN WRITTEN POLICIES OF THE CORPORATION.

CONTENTS

Contributors

ASSA ABLOY	Patrik Björling Rygert
ASSA ABLOY	Mattias Rengstedt
Axis Communications AB	Robert Rosengren
Axis Communications AB	Derek Wang
Axis Communications AB	Emil Selinder
Bosch	Dirk Schreiber
Honeywell	Uvaraj Thangarajan
Honeywell	Vinay Ghule
Siemens AG	Lokeshwar K
Siemens AG	Suresh Raman
Siemens AG	Suresh Krishnamurthy

1 Scope

1.1 General

This specification defines the web service interface for interaction with ONVIF devices which support scheduled authentication behavior for access points.

Web service usage and common ONVIF functionality are outside the scope of this document. Please refer to [Core Specification] for more information.

1.2 Conventions

The key words “shall”, “shall not”, “should”, “should not”, “may”, “need not”, “can”, “cannot” in this specification are to be interpreted as described in Annex H of [ISO/IEC Directives].

1.3 Namespaces

This document references the following namespaces:

Table 1 – Referenced namespaces (with prefix)

Prefix	Namespace URI
env	http://www.w3.org/2003/05/soap-envelope
ter	http://www.onvif.org/ver10/error
xs	http://www.w3.org/2001/XMLSchema
tt	http://www.onvif.org/ver10/schema
pt	http://www.onvif.org/ver10/pacs
tns1	http://www.onvif.org/ver10/topics
tab	http://www.onvif.org/ver10/authenticationbehavior/wsdl

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ONVIF Core Specification
<http://www.onvif.org/specs/core/ONVIF-Core-Specification.pdf>

ONVIF PACS Architecture and Design Considerations
<https://www.onvif.org/specs/wp/ONVIF-PACS-Architecture-and-Design-Considerations.pdf>

ONVIF Schedule Service Specification
<http://www.onvif.org/specs/srv/sched/ONVIF-Scheduler-Service-Spec.pdf>

ISO/IEC Directives, *ISO/IEC Directives Part 2, Principles and rules for the structure and drafting of ISO and IEC documents, Edition 7.0, May 2016*
<http://www.iec.ch/members_experts/refdocs/iec/isoiecdir-2%7Bed7.0%7Den.pdf>

3 Terms, definitions and abbreviations

3.1 Terms and definitions

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

Authentication Policy Each authentication policy associates a security level with a schedule (during which the specified security level will be required at the access point).

Authentication Profile Authentication profiles are used to define authentication behavior for a type of access points. For instance, all entrance access points are configured to require Card access during office hours, Card+PIN access during nighttime, and no access during holidays.

Recognition Recognition is the action of identifying authorized users requesting access by the comparison of presented credential data with recorded credential data.

Recognition Group Recognition groups are used to define a logical OR between the recognition methods in a security level.

Example: One recognition group contains the recognition methods pt:Card and pt:Fingerprint. Another group contains the recognition methods pt:Card and pt:Face. The resulting effect is that the access point will require either Card+Fingerprint, or Card+Face.

Recognition Method A recognition method is either memorized, biometric or held within a physical credential.

Recognition Type A recognition type is either a recognition method or a physical input such as a request-to-exit button.

Security Level Security Levels are defined as individual recognition methods, combinations of recognition methods (using logical AND or OR), or no recognition methods (open). Security levels are given explanatory names, such as “Card”, “Card+ PIN”, “Fingerprint or Iris”, “Open”, etc.

3.2 Abbreviated terms

PACS Physical Access Control System

4 Overview

4.1 General

This service offers commands to manage authentication behavior and security levels.

Authentication profiles are used to define how credential holders can be granted access to an access point by defining when different security levels are required.

4.2 Example

The following example uses a schedule (see [ONVIF Schedule Service Specification]) that defines the following:

- Time range of 9 AM-5 PM during regular Mondays-Fridays
- No time ranges are defined for regular Saturdays and Sundays
- A special day group for half-working days, with a time range of 9 AM-1 PM
- A special day group for bank holidays, with a time range of 9 AM-5 PM

Additionally, this service defines four security levels; “Card”, “Card+PIN”, “Dual Card” and “No access” (see section 5.3.2.2).

By using security level constraints (see section 5.2.2.4), we can map the four different schedule states to a security level and an authentication mode:

Special Day?	Time range active?	Resulting security level	Resulting authentication mode
No / Regular day	No	“Card+PIN”	pt:SingleCredential
No / Regular day	Yes	“Card”	pt:SingleCredential
“Half-working days”	No	“Card+PIN”	pt:SingleCredential
“Half-working days”	Yes	“Card”	pt:SingleCredential
“Bank holidays”	No	“No access”	(irrelevant)
“Bank holidays”	Yes	“Card”	pt:DualCredential

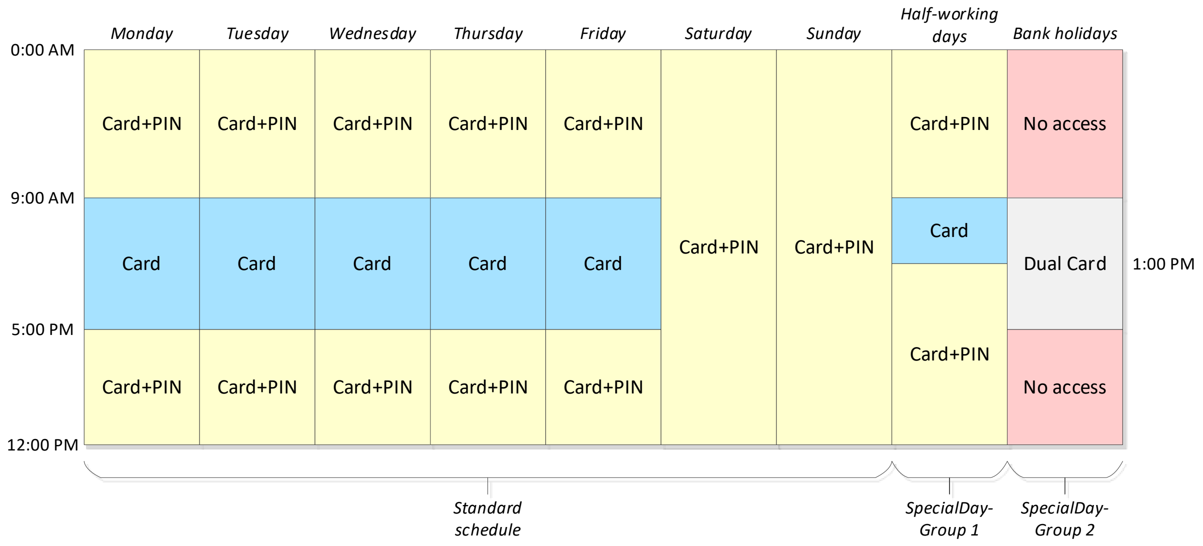

Figure 1 – Authentication behavior example

5 Authentication behavior

5.1 Service capabilities

5.1.1 General

The device shall provide service capabilities in two ways:

1. With the GetServices method of Device service when IncludeCapability is true. Please refer to [Core Specification] for more details.

2. With the GetServiceCapabilities method.

5.1.2 Data structures

5.1.2.1 ServiceCapabilities

The service capabilities reflect optional functionality of a service. The information is static and does not change during device operation. The following capabilities are available:

- **MaxLimit**

  The maximum number of entries returned by a single Get<Entity>List or Get<Entity> request. The device shall never return more than this number of entities in a single response.

- **MaxAuthenticationProfiles**

  Indicates the maximum number of authentication profiles the device supports. The device shall support at least one authentication profile.

- **MaxPoliciesPerAuthenticationProfile**

  Indicates the maximum number of authentication policies per authentication profile supported by the device.

- **MaxSecurityLevels**

  Indicates the maximum number of security levels the device supports. The device shall support at least one security level.

- **MaxRecognitionGroupsPerSecurityLevel**

  Indicates the maximum number of recognition groups per security level supported by the device.

- **MaxRecognitionMethodsPerRecognitionGroup**

  Indicates the maximum number of recognition methods per recognition group supported by the device.

- **ClientSuppliedTokenSupported**

  Indicates that the client is allowed to supply the token when creating authentication profiles and security levels. To enable the use of the commands SetAuthenticationProfile and SetSecurityLevel, the value must be set to true.

- **SupportedAuthenticationModes**

  A list of supported authentication modes (including custom modes). See section 5.2.2.4 (AuthenticationMode field) for supported authentication modes. This field is optional, and when omitted, the client shall assume that the device supports “pt:SingleCredential” only.

5.1.3 GetServiceCapabilities command

This operation returns the capabilities of the authentication behavior service.

Table 2 GetServiceCapabilities command

GetServiceCapabilities	Access Class: PRE_AUTH
Message name	**Description**
GetServiceCapabilitiesRequest	*This message shall be empty*
GetServiceCapabilitiesResponse	*This message contains:* • *"Capabilities": The capability response message contains the requested authentication behavior service capabilities using a hierarchical XML capability structure.* tab:ServiceCapabilities **Capabilities [1][1]**

5.2 Authentication profile information

5.2.1 General

Authentication profiles are used to define authentication behavior for a type of access points. For instance, all entrance access points are configured to require Card access during office hours, Card+PIN access during nighttime, and no access during holidays.

The authentication behavior of an access point type is defined by associating security levels with schedules. When the schedule is active, the specified security level is required.

If a certain point in time is not covered by any schedule, then the access point is set to the default security level.

The following figure shows an overview of the related objects of an authentication profile:

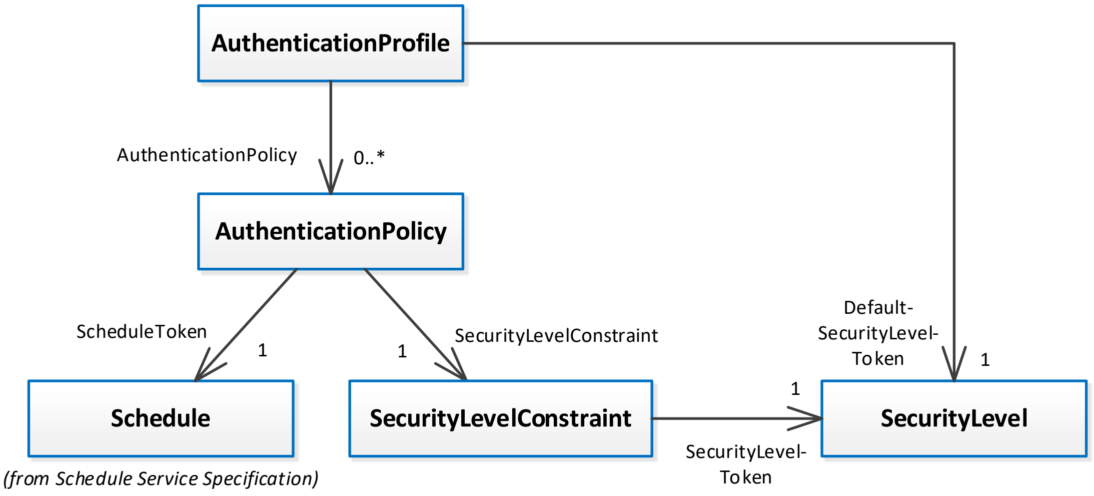

Figure 2 – The related objects of an authentication profile

5.2.2 Data structures

5.2.2.1 AuthenticationProfileInfo

The AuthenticationProfileInfo structure contains information of a specific authentication profile instance.

The device shall provide the following fields for each AuthenticationProfileInfo instance:

- **token**

  A service unique identifier of the authentication profile.

- **Name**

  A descriptive name, such as “Entrance doors - entry”. It shall be up to 64 characters.

To provide more information, the device may include the following optional fields:

- **Description**

  User readable description for the authentication profile. It shall be up to 1024 characters.

  Note that when this optional field is omitted, the device will assume an empty value.

5.2.2.2 AuthenticationProfile

The AuthenticationProfile structure shall include all properties of the AuthenticationProfileInfo structure and also a default security level, an authentication mode, and a list of AuthenticationPolicy instances.

The device shall provide the following fields for each AuthenticationProfile instance:

- **DefaultSecurityLevelToken**

  The default security level is used if none of the authentication policies has a schedule covering the time of access (or if no authentication policies are defined).

- **AuthenticationPolicy**

  Each authentication policy associates a security level with a schedule (during which the specified security level will be required at the access point). If no authentication policies are specified, then DefaultSecurityLevelToken will be used.

  Note that when an authentication profile is updated, then any previous authentication policies are replaced with the new list.

5.2.2.3 AuthenticationPolicy

The authentication policy is an association of a security level and a schedule. It defines when a certain security level is required to grant access to a credential holder. Each security level is given a unique priority. If authentication policies have overlapping schedules, the security level with the highest priority is used.

The device shall provide the following fields for each authentication policy instance:

- **ScheduleToken**

  Reference to the schedule used by the authentication policy. Schedules are defined in [ONVIF Schedule Service Specification].

- **SecurityLevelConstraint**

  A list of security level constraint structures defining the conditions for what security level to use.

  Minimum one security level constraint must be specified.

Note that when an authentication policy is updated, then any previous security level constraints are replaced with the new list.

5.2.2.4 SecurityLevelConstraint

This structure defines what security level should be active depending on the state of the schedule. The state of a schedule has two boolean values corresponding to four different states:

- The standard schedule is active, and it is currently no special day
- The standard schedule is inactive, and it is currently no special day
- It is a special day, and a time period defined in the special days schedule is active
- It is a special day, but no time periods defined in the special days schedule are active

If the state of the schedule corresponds to the ActiveRegularSchedule and ActiveSpecialDaySchedule settings in this structure, then the specified security level will be used.

Please note that if the device do not support special days, the value of the field ActiveSpecialDaySchedule will be ignored.

The device shall provide the following fields for each security level constraint instance:

- **ActiveRegularSchedule**

  Corresponds to the Active field in the ScheduleState structure in [ONVIF Schedule Service Specification].

- **ActiveSpecialDaySchedule**

  Corresponds to the SpecialDay field in the ScheduleState structure in [ONVIF Schedule Service Specification].

  This field will be ignored if the device do not support special days.

- **AuthenticationMode**

  Defines the mode of authentication. Authentication modes starting with the prefix pt: are reserved to define ONVIF-specific authentication modes. For custom defined authentication modes, free text can be used. The following authentication modes are defined by ONVIF:

  - pt:SingleCredential — Normal mode where only one credential holder is required to be granted access.
  - pt:DualCredential — Two credential holders are required to be granted access.

  This field is optional, and if omitted, the default value “pt:SingleCredential” is assumed.

- **SecurityLevelToken**

  Reference to the security level used by the authentication policy.

5.2.3 GetAuthenticationProfileInfo command

This operation requests a list of AuthenticationProfileInfo items matching the given tokens.

The device shall ignore tokens it cannot resolve and shall return an empty list if there are no items matching the specified tokens. The device shall not return a fault in this case.

If the number of requested items is greater than MaxLimit, a TooManyItems fault shall be returned.

Table 3 GetAuthenticationProfileInfo command

GetAuthenticationProfileInfo	Access Class: READ_SYSTEM
Message name	**Description**
GetAuthenticationProfileInfoRequest	*This message contains:* • "*Token": Tokens of AuthenticationProfileInfo items to get.* pt:ReferenceToken **Token [1][unbounded]**
GetAuthenticationProfileInfoResponse	*This message contains:* • *"AuthenticationProfileInfo": List of AuthenticationProfileInfo items.* tab:AuthenticationProfileInfo **AuthenticationProfileInfo [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgs ter:TooManyItems	*Too many items were requested, see MaxLimit capability.*

5.2.4 GetAuthenticationProfileInfoList command

This operation requests a list of all AuthenticationProfileInfo items provided by the device.

A call to this method shall return a StartReference when not all data is returned and more data is available. The reference shall be valid for retrieving the next set of data. Please refer to section 4.8.3 in [ONVIF PACS Architecture and Design Considerations] for more details.

The number of items returned shall not be greater than the Limit parameter.

Table 4 GetAuthenticationProfileInfoList command

GetAuthenticationProfileInfoList	Access Class: READ_SYSTEM
Message name	**Description**
GetAuthenticationProfileInfoListRequest	*This message contains:* • *"Limit": Maximum number of entries to return. If not specified, less than one or higher than what the device supports, the number of items is determined by the device.* • *"StartReference": Start returning entries from this start reference. If not specified, entries shall start from the beginning of the dataset.* xs:int **Limit [0][1]** xs:string **StartReference [0][1]**
GetAuthenticationProfileInfoListResponse	*This message contains:* • *"NextStartReference": StartReference to use in next call to get the following items. If absent, no more items to get.* • *"AuthenticationProfileInfo": List of AuthenticationProfileInfo items.* xs:string **NextStartReference [0][1]** tab:AuthenticationProfileInfo **AuthenticationProfileInfo [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:InvalidStartReference	*StartReference is invalid or has timed out. Client needs to start fetching from the beginning.*

5.2.5 GetAuthenticationProfiles command

This operation requests a list of AuthenticationProfile item matching the given tokens.

The device shall ignore tokens it cannot resolve and shall return an empty list if there are no items matching the specified tokens. The device shall not return a fault in this case.

If the number of requested items is greater than MaxLimit, a TooManyItems fault shall be returned.

Table 5 GetAuthenticationProfiles command

GetAuthenticationProfiles	Access Class: READ_SYSTEM
Message name	**Description**
GetAuthenticationProfileRequest	*This message contains:* • "*Token": Tokens of AuthenticationProfile items to get* pt:ReferenceToken **Token [1][unbounded]**
GetAuthenticationProfileResponse	*This message contains:* • *"AuthenticationProfile": List of AuthenticationProfile items.* tab:AuthenticationProfile **AuthenticationProfile [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgs ter:TooManyItems	*Too many items were requested, see MaxLimit capability.*

5.2.6 GetAuthenticationProfileList command

This operation requests a list of all AuthenticationProfile items provided by the device.

A call to this method shall return a StartReference when not all data is returned and more data is available. The reference shall be valid for retrieving the next set of data. Please refer to section 4.8.3 in [ONVIF PACS Architecture and Design Considerations] for more details.

The number of items returned shall not be greater than the Limit parameter.

Table 6 GetAuthenticationProfileList command

GetAuthenticationProfileList	Access Class: READ_SYSTEM
Message name	**Description**
GetAuthenticationProfileListRequest	*This message contains:* • *"Limit": Maximum number of entries to return. If not specified, less than one or higher than what the device supports, the number of items is determined by the device.* • *"StartReference": Start returning entries from this start reference. If not specified, entries shall start from the beginning of the dataset.* xs:int **Limit [0][1]** xs:string **StartReference [0][1]**
GetAuthenticationProfileListResponse	*This message contains:* • *"NextStartReference": StartReference to use in next call to get the following items. If absent, no more items to get.* • *"AuthenticationProfile": List of AuthenticationProfile items.* xs:string **NextStartReference [0][1]** tab:AuthenticationProfile **AuthenticationProfile [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:InvalidStartReference	*StartReference is invalid or has timed out. Client needs to start fetching from the beginning.*

5.2.7 CreateAuthenticationProfile command

This operation creates the specified authentication profile in the device.

The token field of the AuthenticationProfile structure shall be empty and the device shall allocate a token for the authentication profile. The allocated token shall be returned in the response.

If the client sends any value in the token field, the device shall return InvalidArgVal as a generic fault code.

Table 7 CreateAuthenticationProfile command

CreateAuthenticationProfile	Access Class: WRITE_SYSTEM
Message name	**Description**
CreateAuthenticationProfileRequest	*This message contains:* • *"AuthenticationProfile": The authentication profile to create* tab:AuthenticationProfile **AuthenticationProfile [1][1]**
CreateAuthenticationProfileResponse	*This message contains:* • *"Token": The token of the created authentication profile* pt:ReferenceToken **Token [1][1]**
Fault codes	**Description**
env:Receiver ter:CapabilityViolated ter:MaxAuthenticationProfiles	*There is not enough space to add a new authentication profile, see MaxAuthenticationProfiles capability*
env:Sender ter:CapabilityViolated ter:MaxPoliciesPerAuthenticationProfile	*There are too many AuthenticationPolicy entities referred in this AuthenticationProfile, see MaxPoliciesPerAuthenticationProfile capability.*
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	*A referred entity token is not found (some devices may not validate referred entities).*

5.2.8 SetAuthenticationProfile command

This method is used to synchronize an authentication profile in a client with the device.

If an authentication profile with the specified token does not exist in the device, the authentication profile is created. If an authentication profile with the specified token exists, then the authentication profile is modified.

A call to this method takes an AuthenticationProfile structure as input parameter. The token field of the AuthenticationProfile shall not be empty.

A device that signals support for the ClientSuppliedTokenSupported capability shall implement this command.

If no token was specified in the request, the device shall return InvalidArgs as a generic fault code.

Table 8 SetAuthenticationProfile command

SetAuthenticationProfile	Access Class: WRITE_SYSTEM
Message name	**Description**
SetAuthenticationProfileRequest	*This message contains:* • *"AuthenticationProfile": The AuthenticationProfile to create or modify* tab:AuthenticationProfile **AuthenticationProfile [1][1]**
SetAuthenticationProfileResponse	*This message shall be empty*
Fault codes	**Description**
env:Receiver ter:CapabilityViolated ter:ClientSuppliedTokenSupported	*The device does not support that the client supplies the token*
env:Receiver ter:CapabilityViolated ter:MaxAuthenticationProfiles	*There is not enough space to add new AuthenticationProfile, see MaxAuthenticationProfiles capability*
env:Sender ter:CapabilityViolated ter:MaxPoliciesPerAuthenticationProfile	*There are too many AuthenticationPolicy entities referred in this AuthenticationProfile, see MaxPoliciesPerAuthenticationProfile capability.*
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	*A referred entity token is not found (some devices may not validate referred entities).*
env:Sender ter:CapabilityViolated ter:SupportedSecurityLevels	*The updated authentication profile contains security levels that is not supported by an access point that is referring to this authentication profile (some devices may not validate this).*

5.2.9 ModifyAuthenticationProfile command

This operation modifies the specified authentication profile.

The token of the authentication profile to modify is specified in the token field of the AuthenticationProfile structure and shall not be empty. All other fields in the structure shall overwrite the fields in the specified authentication profile.

If no token was specified in the request, the device shall return InvalidArgs as a generic fault code.

Table 9 ModifyAuthenticationProfile command

ModifyAuthenticationProfile	Access Class: WRITE_SYSTEM
Message name	**Description**
ModifyAuthenticationProfileRequest	*This message contains:* • *"AuthenticationProfile": The AuthenticationProfile to modify* tab:AuthenticationProfile **AuthenticationProfile [1][1]**
ModifyAuthenticationProfileResponse	*This message shall be empty*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*The specified token not found.*
env:Sender ter:CapabilityViolated ter:MaxPoliciesPerAuthenticationProfile	*There are too many AuthenticationPolicy entities referred in this AuthenticationProfile, see MaxPoliciesPerAuthenticationProfile capability.*
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	*A referred entity token is not found (some devices may not validate referred entities).*
env:Sender ter:CapabilityViolated ter:SupportedSecurityLevels	*The updated authentication profile contains security levels that is not supported by an access point that is referring to this authentication profile (some devices may not validate this).*

5.2.10 DeleteAuthenticationProfile command

This operation deletes the specified authentication profile.

If the authentication profile is deleted, all authentication policies associated with the authentication profile will also be deleted.

If it is associated with one or more entities some devices may not be able to delete the authentication profile, and consequently a ReferenceInUse fault shall be generated.

If no token was specified in the request, the device shall return InvalidArgs as a generic fault code.

Table 10 DeleteAuthenticationProfile command

DeleteAuthenticationProfile	Access Class: WRITE_SYSTEM
Message name	**Description**
DeleteAuthenticationProfileRequest	*This message contains:* • *"Token": The token of the AuthenticationProfile to delete.* pt:ReferenceToken **Token [1][1]**
DeleteAuthenticationProfileResponse	*This message shall be empty*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*AuthenticationProfile token is not found.*
env:Sender ter:InvalidArgVal ter:ReferenceInUse	*Failed to delete, AuthenticationProfile token is in use*

5.3 Security level information

5.3.1 General

Security Levels are defined as individual recognition methods, combinations of recognition methods (using logical AND or OR), or no recognition methods (open). Security levels are given explanatory names, such as “Card”, “Card+ PIN”, “Fingerprint or Iris”, “Open”, etc.

Possible authentication factors are Card, PIN, Fingerprint, Face, Iris, Vein, Palm and REX (that is not hardwired to directly unlock, but where the device can do some action, such as taking a decision).

The following figure shows an overview of the related objects of a security level:

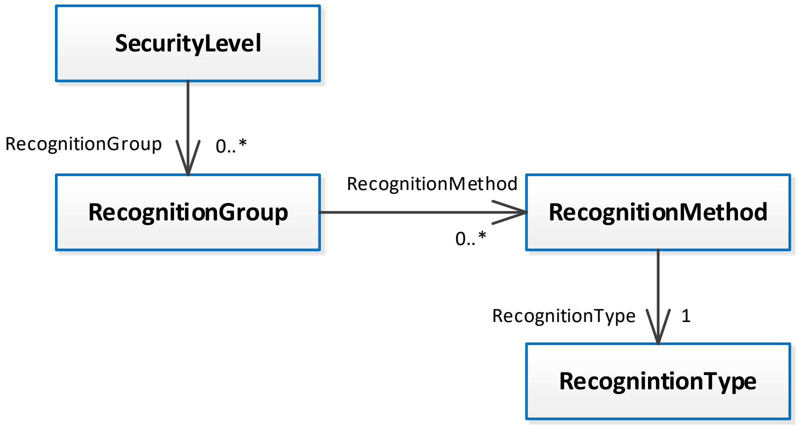

Figure 3 – The related objects of a security level

5.3.2 Data structures

5.3.2.1 SecurityLevelInfo

The SecurityLevelInfo structure contains information of a specific security level instance.

The device shall provide the following fields for each SecurityLevelInfo instance:

- **token**

  A service-unique identifier of the security level.

- **Name**

  User readable name. It shall be up to 64 characters.

- **Priority**

  A higher number indicates that the security level is considered more secure than security levels with lower priorities. The priority is used when an authentication profile have overlapping schedules with different security levels. When an access point is accessed, the authentication policies are walked through in priority order (highest priority first). When a schedule is found covering the time of access, the associated security level is used and processing stops. Two security levels cannot have the same priority.

To provide more information, the device may include the following optional fields:

- **Description**

  User readable description for the special days. It shall be up to 1024 characters.

  Note that when this optional field is omitted, the device will assume an empty value.

5.3.2.2 SecurityLevel

The SecurityLevel structure shall include all properties of the SecurityLevelInfo structure and also a set of recognition groups.

The recognition groups are used to define a logical OR between the groups. Each recognition group consists of one or more recognition methods.

Example: One recognition group contains the recognition methods pt:Card and pt:Fingerprint. Another group contains the recognition methods pt:Card and pt:Face. The resulting effect is that the access point will require either Card+Fingerprint, or Card+Face.

The order of the requested recognition methods can be defined at the access point using the order field in the recognition method structure.

The following example shows three different security levels. The first has no recognition groups and the access point will not require any authentication when active. The second is described above. The third defines one recognition group but with no recognition methods, which has the same effect as if the access point is disabled.

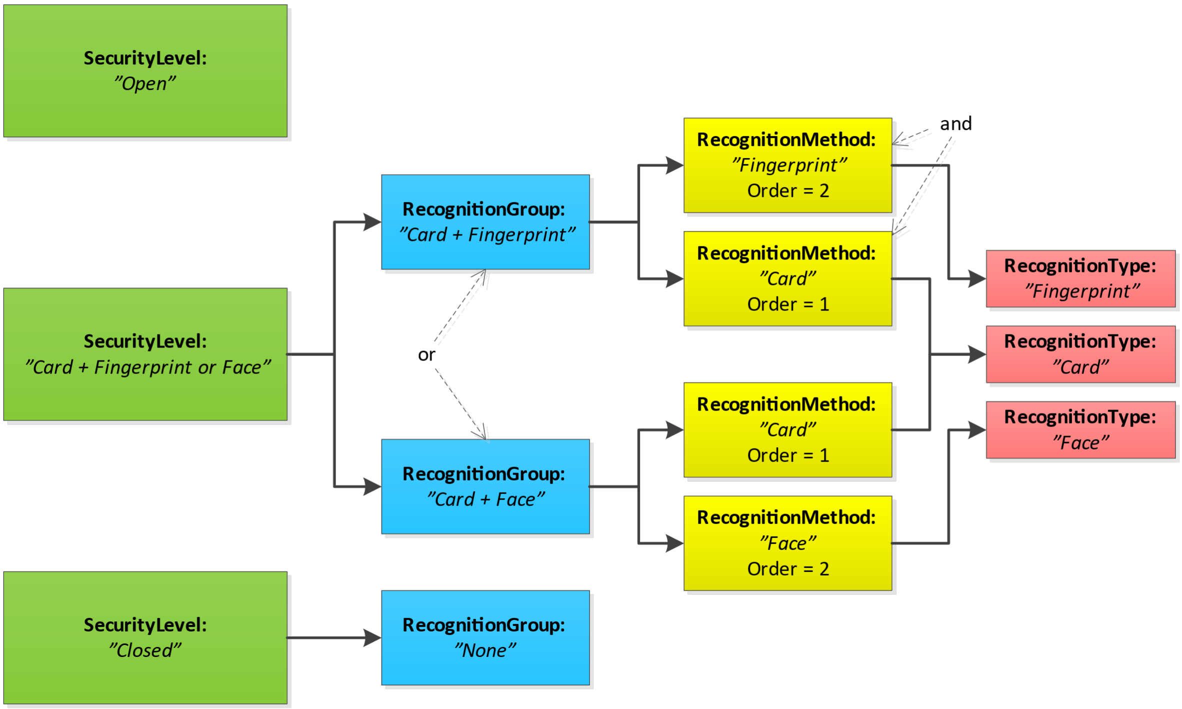

Figure 4 – Security level examples

The device shall provide the following fields for each SecurityLevel instance:

- **RecognitionGroup**

  The recognition groups are used to define a logical OR between the groups. Each recognition group consists of one or more recognition methods.

  No recognition groups mean that the access point is open.

  Note that when a security level is updated, then any previous recognition groups are replaced with the new list.

5.3.2.3 RecognitionGroup

The device shall provide the following fields for each RecognitionGroup instance:

- **RecognitionMethod**

A list of recognition methods to request for at the access point. No recognition methods mean that the access point is closed.

Note that when a recognition group is updated, then any previous recognition methods are replaced with the new list.

5.3.2.4 RecognitionMethod

Recognition is the action of identifying authorized users requesting access by the comparison of presented credential data with recorded credential data. A recognition method is either memorized, biometric or held within a physical credential. A recognition type is either a recognition method or a physical input such as a request-to-exit button.

The device shall provide the following fields for each RecognitionMethod instance:

- **RecognitionType**

  The requested type of recognition. Is of type text.

  Recognition types starting with the prefix pt: are reserved to define ONVIF-specific types as defined in pt:RecognitionType. For custom defined identifier types, free text can be used.

- **Order**

  The order value defines when this recognition method will be requested in relation to the other recognition methods in the same security level. A lower number indicates that the recognition method will be requested before recognition methods with a higher number.

5.3.3 GetSecurityLevelInfo command

This operation requests a list of SecurityLevelInfo items matching the given tokens.

The device shall ignore tokens it cannot resolve and shall return an empty list if there are no items matching the specified tokens. The device shall not return a fault in this case.

If the number of requested items is greater than MaxLimit, a TooManyItems fault shall be returned.

Table 11 GetSecurityLevelInfo command

GetSecurityLevelInfo	Access Class: READ_SYSTEM
Message name	**Description**
GetSecurityLevelInfoRequest	*This message contains:* • *"Token": Tokens of SecurityLevelInfo items to get.* pt:ReferenceToken **Token [1][unbounded]**
GetSecurityLevelInfoResponse	*This message contains:* • *"SecurityLevelInfo": List of SecurityLevelInfo items.* tab:SecurityLevelInfo **SecurityLevelInfo [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgs ter:TooManyItems	*Too many items were requested, see MaxLimit capability.*

5.3.4 GetSecurityLevelInfoList command

This operation requests a list of all SecurityLevelInfo items provided by the device.

A call to this method shall return a StartReference when not all data is returned and more data is available. The reference shall be valid for retrieving the next set of data. Please refer to section 4.8.3 in [ONVIF PACS Architecture and Design Considerations] for more details.

The number of items returned shall not be greater than the Limit parameter.

Table 12 GetSecurityLevelInfoList command

GetSecurityLevelInfoList	Access Class: READ_SYSTEM
Message name	**Description**
GetSecurityLevelInfoListRequest	*This message contains:* • *"Limit": Maximum number of entries to return. If not specified, less than one or higher than what the device supports, the number of items is determined by the device.* • *"StartReference": Start returning entries from this start reference. If not specified, entries shall start from the beginning of the dataset.* xs:int **Limit [0][1]** xs:string **StartReference [0][1]**
GetSecurityLevelInfoListResponse	*This message contains:* • *"NextStartReference": StartReference to use in next call to get the following items. If absent, no more items to get.* • *"SecurityLevelInfo": List of SecurityLevelInfo items.* xs:string **NextStartReference [0][1]** tab:SecurityLevelInfo **SecurityLevelInfo [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:InvalidStartReference	*StartReference is invalid or has timed out. Client needs to start fetching from the beginning.*

5.3.5 GetSecurityLevels command

This operation requests a list of SecurityLevel items matching the given tokens.

The device shall ignore tokens it cannot resolve and shall return an empty list if there are no items matching specified tokens. The device shall not return a fault in this case.

If the number of requested items is greater than MaxLimit, a TooManyItems fault shall be returned.

Table 13 GetSecurityLevels command

GetSecurityLevels	Access Class: READ_SYSTEM
Message name	**Description**
GetSecurityLevelsRequest	*This message contains:* • "*Token": Tokens of the SecurityLevel items to get.* pt:ReferenceToken **Token [1][unbounded]**
GetSecurityLevelsResponse	*This message contains:* • *"SecurityLevel": List of SecurityLevel items.* tab:SecurityLevel **SecurityLevel [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgs ter:TooManyItems	*Too many items were requested, see MaxLimit capability.*

5.3.6 GetSecurityLevelList command

This operation requests a list of all SecurityLevel items provided by the device.

A call to this method shall return a StartReference when not all data is returned and more data is available. The reference shall be valid for retrieving the next set of data. Please refer to section 4.8.3 in [ONVIF PACS Architecture and Design Considerations] for more details.

The number of items returned shall not be greater than the Limit parameter.

Table 14 GetSecurityLevelList command

GetSecurityLevelList	Access Class: READ_SYSTEM
Message name	**Description**
GetSecurityLevelListRequest	*This message contains:* • *"Limit": Maximum number of entries to return. If not specified, less than one or higher than what the device supports, the number of items is determined by the device.* • *"StartReference": Start returning entries from this start reference. If not specified, entries shall start from the beginning of the dataset.* xs:int **Limit [0][1]** xs:string **StartReference [0][1]**
GetSecurityLevelListResponse	*This message contains:* • *"NextStartReference": StartReference to use in next call to get the following items. If absent, no more items to get.* • *"SecurityLevel": List of SecurityLevel items.* xs:string **NextStartReference [0][1]** tab:SecurityLevel **SecurityLevel [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:InvalidStartReference	*StartReference is invalid or has timed out. Client needs to start fetching from the beginning.*

5.3.7 CreateSecurityLevel command

This operation creates the specified security level in the device.

The token field of the SecurityLevel structure shall be empty and the device shall allocate a token for the security level. The allocated token shall be returned in the response.

If the client sends any value in the token field, the device shall return InvalidArgVal as a generic fault code.

Table 15 CreateSecurityLevel command

CreateSecurityLevel	Access Class: WRITE_SYSTEM
Message name	**Description**
CreateSecurityLevelRequest	*This message contains:* • *"SecurityLevel": The security level to create.* tab:SecurityLevel **SecurityLevel [1][1]**
CreateSecurityLevelResponse	*This message contains:* • *"Token": The token of created security level.* pt:ReferenceToken **Token [1][1]**
Fault codes	**Description**
env:Receiver ter:CapabilityViolated ter:MaxSecurityLevels	*There is not enough space to add new SecurityLevel items, see the MaxSecurityLevels capability*
env:Sender ter:CapabilityViolated ter:MaxRecognitionGroupsPerSecurityLevel	*There are too many recognition method groups in a SecurityLevel, see MaxRecognitionGroupsPer-SecurityLevel capability.*
env:Sender ter:CapabilityViolated ter:MaxRecognitionMethodsPerRecognition-Group	*There are too many recognition methods in a recognition group, see MaxRecognition-MethodsPerRecognitionGroup capability.*
env:Sender ter:InvalidArgVal ter:DuplicatePriority	*The priority of the security level is already used by another security level.*

5.3.8 SetSecurityLevel command

This method is used to synchronize a security level in a client with the device.

If a security level with the specified token does not exist in the device, the security level is created. If a security level with the specified token exists, then the security level is modified.

A call to this method takes a SecurityLevel structure as input parameter. The token field of the SecurityLevel shall not be empty.

A device that signals support for the ClientSuppliedTokenSupported capability shall implement this command.

Table 16 SetSecurityLevel command

SetSecurityLevel	Access Class: WRITE_SYSTEM
Message name	**Description**
SetSecurityLevelRequest	*This message contains:* - *"SecurityLevel": The security level to create or modify.* tab:SecurityLevel **SecurityLevel [1][1]**
SetSecurityLevelResponse	*This message shall be empty*
Fault codes	**Description**
env:Receiver ter:CapabilityViolated ter:ClientSuppliedTokenSupported	*The device does not support that the client supplies the token*
env:Receiver ter:CapabilityViolated ter:MaxSecurityLevels	*There is not enough space to add new SecurityLevel items, see the MaxSecurityLevels capability*
env:Sender ter:CapabilityViolated ter:MaxRecognitionGroupsPerSecurityLevel	*There are too many recognition method groups in a SecurityLevel, see MaxRecognitionGroupsPer-SecurityLevel capability.*
env:Sender ter:CapabilityViolated ter:MaxRecognitionMethodsPerRecognition-Group	*There are too many recognition methods in a recognition method group, see MaxRecognition-MethodsPerRecognitionGroup capability.*
env:Sender ter:InvalidArgVal ter:DuplicatePriority	*The priority of the security level is already used by another security level.*
env:Sender ter:InvalidArgVal ter:MissingToken	*The token of the security level item must be specified.*

5.3.9 ModifySecurityLevel command

This operation modifies the specified security level.

The token of the security level to modify is specified in the token field of the SecurityLevel structure and shall not be empty. All other fields in the structure shall overwrite the fields in the specified security level.

Table 17 ModifySecurityLevel command

ModifySecurityLevel	Access Class: WRITE_SYSTEM
Message name	**Description**
ModifySecurityLevelRequest	*This message contains:* - *"SecurityLevel": The security level to modify.* tab:SecurityLevel **SecurityLevel [1][1]**
ModifySecurityLevelResponse	*This message shall be empty*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*The specified token not found.*
env:Sender ter:CapabilityViolated ter:MaxRecognitionGroupsPerSecurityLevel	*There are too many recognition method groups in a SecurityLevel, see MaxRecognitionGroupsPer-SecurityLevel capability.*
env:Sender ter:CapabilityViolated ter:MaxRecognitionMethodsPerRecognition-Group	*There are too many recognition methods in a recognition method group, see MaxRecognition-MethodsPerRecognitionGroup capability.*
env:Sender ter:InvalidArgVal ter:DuplicatePriority	*The priority of the security level is already used by another security level.*

5.3.10 DeleteSecurityLevel command

This method deletes the specified security level.

If it is associated with one or more entities some devices may not be able to delete the security level, and consequently a ReferenceInUse fault shall be generated.

Table 18 DeleteSecurityLevel command

DeleteSecurityLevel	Access Class: WRITE_SYSTEM
Message name	**Description**
DeleteSecurityLevelRequest	*This message contains:* • *"Token": The token of the security level item to delete.* pt:ReferenceToken **Token [1][1]**
DeleteSecurityLevelResponse	*This message shall be empty*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Security level token is not found.*
env:Sender ter:InvalidArgVal ter:ReferenceInUse	*Failed to delete, security level token is in use*

6 Notification topics

6.1 General

This section defines notification topics specific to the authentication behavior service.

6.2 Event overview (informative)

The authentication behavior service specifies events when authentication profiles or security levels are changed.

The main topics for configuration change notifications are:

- tns1:Configuration/AuthenticationProfile/Changed
- tns1:Configuration/AuthenticationProfile/Removed
- tns1:Configuration/SecurityLevel/Changed
- tns1:Configuration/SecurityLevel/Removed

6.3 Configuration changes

6.3.1 General

Whenever configuration data has been changed, added or been removed, the device shall provide these events to inform subscribed clients.

6.3.2 Authentication profile

Whenever configuration data for an authentication profile is changed or an authentication profile is added, the device shall provide the following event:

```
Topic: tns1:Configuration/AuthenticationProfile/Changed

<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="AuthenticationProfileToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
</tt:MessageDescription>
```

Whenever an authentication profile is removed, the device shall provide the following event:

```
Topic: tns1:Configuration/AuthenticationProfile/Removed

<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="AuthenticationProfileToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
</tt:MessageDescription>
```

6.3.3 Security level

Whenever configuration data for a security level is changed or a security level is added, the device shall provide the following event:

```
Topic: tns1:Configuration/SecurityLevel/Changed

<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="SecurityLevelToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
</tt:MessageDescription>
```

Whenever a security level is removed, the device shall provide the following event:

```
Topic: tns1:Configuration/SecurityLevel/Removed

<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="SecurityLevelToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
</tt:MessageDescription>
```

Annex A. Revision History

Rev.	Date	Editor	Changes
18.06	Jun-2018	Patrik Björling Rygert	Initial version
18.12	Dec-2018	Hiroyuki Sano	Change Request 2391, 2393
19.06	Jun-2019	Hiroyuki Sano	Change Request 2457, 2475
19.12	Dec-2019	Hiroyuki Sano	Change Request 2471