Attestation Service for Intel® Software Guard Extensions (Intel® SGX): API Documentation

Revision: 3.0

1 Abbreviations 4
2 Attestation Service for Intel® SGX 5
2.1 Supported environments 5
2.2 Authentication 5
2.2.1 Supported TLS Versions 5
2.3 Available API versions 5
2.3.1 Summary of API v2 changes 5
2.4 Registering for the Service 6
2.5 Troubleshooting 6
3 Attestation API (version 2) 7
3.1 Retrieve SigRL 7
3.1.1 Description 7
3.1.2 API Details 7
3.1.3 Examples 8
3.1.3.1 SigRL Exists 8
3.1.3.2 SigRL Does Not Exist 8
3.1.3.3 Invalid EPID Group 8
3.2 Verify Attestation Evidence 10
3.2.1 Description 10
3.2.2 API Details 10
3.2.3 Examples 11
3.2.3.1 Without PSE Manifest 11
3.2.3.2 With PSE Manifest 12
3.2.3.3 Quote with Linkable EPID Signature 12
3.2.3.4 With Invalid PSE Manifest 13
3.2.3.5 With Nonce 13
3.2.3.6 With Invalid Quote 14
3.2.3.7 Revoked EPID Group 14
4 Data Structures 16
4.1 Attestation Evidence Payload 16
4.2 Attestation Verification Report 16

4.2.1 Report Data 16
4.2.2 Report Signature 19
4.2.3 Report Signing Certificate Chain 19
4.2.4 Platform Info Blob 20
4.2.4.1 Platform Info Blob TLV 20
4.3 Quoting Data Structures 20
4.3.1 QUOTE Structure 20
4.4 SGX Platform Service Security Property Descriptor 21

1 Abbreviations

Abbreviation	Description
IAS	Attestation Service for Intel® SGX
CA	Certificate Authority
EOL	End of Life
EPID	Enhanced Privacy ID
JSON	JavaScript Object Notation
MTLS	Mutual Transport Layer Security
QE	Quoting Enclave
REST	Representational State Transfer
SP	Service Provider
TCB	Trusted Computing Base
TLV	Type-length-value
UUID	Universally Unique Identifier
{*variable*}	Denotes a variable parameter in the API

2 Attestation Service for Intel® SGX

Attestation Service for Intel® SGX (IAS) is a web service hosted and operated by Intel in a cloud environment. The primary responsibility of the IAS is verification of attestation evidence submitted by Service Providers (SPs).

2.1 Supported environments

Development Environment *– test environment established for software development purposes (early developer integration):*

https://test-as.sgx.trustedservices.intel.com:443

Production Environment *– production-quality environment to be used by production ready software:*

https://as.sgx.trustedservices.intel.com:443

2.2 Authentication

Attestation Service for Intel® SGX uses MTLS (Mutual Transport Layer Security) as an authentication mechanism. This means that both server and client must introduce themselves with valid certificates, both signed by a trusted certificate authority. The client's certificate must be registered in IAS as part of the Service Provider registration process.

2.2.1 Supported TLS Versions

Attestation Service for Intel® SGX only accepts connections protected by TLS 1.2 or higher. IAS will drop any incoming connections utilizing SSL protocol in any version.

2.3 Available API versions

The latest available API version exposed by Attestation Service for Intel® SGX is version 2. Attestation API version 1 is considered deprecated and is on the path to End of Life (EOL) by end of year 2017. This document focuses only on API version 2, for API version 1 refer to the previous revisions of IAS API specification.

2.3.1 Summary of API v2 changes

The changes introduced in Attestation API version 2 are mainly focused in the following areas:

1) Verify Attestation Evidence API has been updated, specifically a new structure of Attestation Verification Report has been introduced, which includes additional fields (isvEnclaveQuoteBody, pseManifestHash) and a simplified pseManifestStatus (see Section 3.2.2 and Section 4.2.1 for further details).
2) Report Signing Certificate Chain has been introduced and it is now returned along with Attestation Verification Report. The certificate chain can be used to verify the digital signature of the Attestation Verification Report instead of using public part of Report Signing Key directly (see Section 4.2.2 and Section 4.2.3 for further details).
3) Attestation Verification Reports are no longer cached in IAS. Version 2 of the Verify Attestation Evidence API does not store generated reports thus Retrieve Attestation Verification Report API that allowed the retrieval of cached reports has been removed.

2.4 Registering for the Service

Registration of Service Providers (SPs) will be handled via the request form linked from http://software.intel.com/sgx. As part of this process, the following artifacts must be delivered by SPs:

- ***X.509 client certificate*** that identifies the SP. This certificate will be registered in IAS and used by the SP to authenticate to the service. As a result IAS will be configured to support only registered SPs. The certificate needs to be issued by a commonly trusted Certificate Authority (CA) (e.g., Verisign) to be registered in the Production Environment. The Development Services Environment does not have that restriction - a valid self-signed certificate can be used. Details on client certificate requirements for IAS are available at the following link: https://software.intel.com/en-us/articles/certificate-requirements-for-intel-attestation-services.
- ***Email address(es)*** that will be used to notify the Service Provider about updates and availability of IAS (e.g. planned and unplanned downtimes, limited availability alerts) as well as revocation data updates.
- ***Linkable/Unlinkable EPID signatures policy setting*** that determines if the Service Provider wants to use Linkable or Unlinkable EPID signatures in enclave quotes.

2.5 Troubleshooting

Each HTTP call to the API will result in a response, containing a header called *Request-ID.* The value of *Request-ID* contains a randomly generated Universally Unique Identifier (UUID) that can be used to track an individual HTTP request. In case of an error, the value of this header should be logged by the SP and included in the issue submission so that further troubleshooting is possible.

3 Attestation API (version 2)

The Attestation API exposed by Attestation Service for Intel® SGX is a programming interface for SPs to verify attestation evidence of SGX enabled enclaves. The API is built using industry-standard Representational State Transfer (REST) architectural style and JavaScript Object Notation (JSON) as the data serialization format.

This specification covers only version 2 of Attestation API (**version 1** is considered **deprecated**).

3.1 Retrieve SigRL

3.1.1 Description

Retrieve the Signature Revocation List (SigRL) for a given EPID group.

SPs are able to retrieve Signature Revocation Lists for EPID groups. EPID SigRLs are generated by Intel and stored in the IAS. They are used to check revocation status of the platform and QE.

Hint: As an optimization, the SP can cache a SigRL retrieved from IAS for a given EPID group and continue to use it until the IAS returns SIGRL_VERSION_MISMATCH for isvEnclaveQuoteStatus in a response to Verify Attestation Evidence. SIGRL_VERSION_MISMATCH indicates that there is a new version of SigRL for a given EPID group that must be used.

3.1.2 API Details

Request		
HTTP method	GET	
HTTP resource	/attestation/sgx/v2/sigrl/{gid} *Note: No trailing slash.*	
Request body	N/A	
Request headers	N/A	
Parameters	{gid} – Base 16-encoded representation of the EPID group ID provided by the platform, encoded as a Big Endian integer.	
Response		
HTTP status	**Status code**	**Description**
	200 OK	Operation successful.
	401 Unauthorized	Failed to authenticate or authorize request.

	404 Not Found	{gid} does not refer to a valid EPID group ID.
	500 Internal Server Error	Internal error occurred.
	503 Service Unavailable	Service is currently not able to process the request (due to a temporary overloading or maintenance). This is a temporary state – the same request can be repeated after some time.
Response headers	Request-ID	Random generated identifier for each request.
Response body	Base 64-encoded SigRL for EPID group identified by {gid} parameter. If {gid} refers to a valid EPID group but there is no SigRL for this group, then the response body shall be empty and the value of Content-Length response header shall be equal to 0. In any other case (error) the response body will be empty, HTTP status code will define the problem and Request-ID header will be returned to allow further troubleshooting.	

3.1.3 Examples

Note: The examples below are only to present sample requests and responses that you might expect from Attestation Service for Intel® SGX in different scenarios. They will not work when used with a real instance of IAS.

3.1.3.1 SigRL Exists

HTTP request		
URI	GET https://test-as.sgx.trustedservices.intel.com:443/attestation/sgx/v2/sigrl/00000010	
HTTP response		
Status	200 OK	
Headers	Request-ID	de305d5475b4431badb2eb6b9e546014
Body	AAIADgAAAAEAAAABAAAAAGSf/es1h/XiJeCg7bXmX0S/NUpJ2jmcEJglQUI8VT5sLGU7iMFu3/UTCv9uPA Dal3LhbrQvhBa6+/dWbj8hnsE=	

3.1.3.2 SigRL Does Not Exist

HTTP request		
URI	GET https://test-as.sgx.trustedservices.intel.com:443/attestation/sgx/v2/sigrl/00000020	
HTTP response		
Status	200 OK	
Headers	Request-ID	de305d5475b4431badb2eb6b9e546014
Body	<empty>	

3.1.3.3 Invalid EPID Group

HTTP request	
URI	GET https://test-as.sgx.trustedservices.intel.com:443/attestation/sgx/v2/sigrl/00000030

HTTP response		
Status	404 Not Found	
Headers	Request-ID	de305d5475b4431badb2eb6b9e546014
Body	<empty>	

3.2 Verify Attestation Evidence

3.2.1 Description

Verify submitted attestation evidence and create a new Attestation Verification Report.

The identity of an ISV enclave and the validity of the platform can be verified using Attestation Service for Intel® SGX. The Attestation Service verifies only the validity of the platform. **It is the responsibility of the Service Provider to validate the ISV enclave identity.** As a result of this process, an Attestation Verification Report will be generated and sent back to the SP. The report will include verification results for:

- QUOTE structure generated by the platform for the ISV enclave
- Optional SGX Platform Service Security Property Descriptor provided by the platform

EPID revocation lists generated by Intel, including EPID Group Revocation Lists (GroupRLs), EPID Private Key Revocation Lists (PrivRLs) and EPID Signature Revocation Lists (SigRLs) will be used to check the revocation status of the platform.

In case the Service Provider registered with a linkable EPID signature policy but uses unlinkable EPID signatures (and vice versa), IAS will respond with "400 Bad Request" to Verify Attestation Evidence call.

Optionally, a signed Platform Info Blob Type-Length-Value (TLV) will be generated and included in the report (as defined in Platform Info Blob section). The SP involved in the remote attestation process should forward Platform Info Blob, excluding the TLV header, to ISV SGX application running on the client platform that is being attested. The ISV SGX application can then process the Platform Info Blob using SGX SDK API sgx_report_attestation_status().

3.2.2 API Details

Request		
HTTP method	POST	
HTTP resource	/attestation/sgx/v2/report *Note: No trailing slash.*	

Request body	Attestation Evidence Payload serialized to JSON:
	{
	"isvEnclaveQuote": "<encoded_quote>",
	"pseManifest":
	"<encoded_SGX_Platform_Service_Security_Property_Descriptor><optional>",
	"nonce": "<custom_value_passed_by_caller><optional>"
	}	

Request headers	**Header**	**Value**
	Content-Type	"application/json"

Parameters	N/A	
Response		
HTTP status code	**Status code**	**Description**
	200 OK	Operation successful.
	400 Bad Request	Invalid Attestation Evidence Payload. The client should not repeat the request without modifications.
	401 Unauthorized	Failed to authenticate or authorize request.
	500 Internal Server Error	Internal error occurred.
	503 Service Unavailable	Service is currently not able to process the request (due to a temporary overloading or maintenance). This is a temporary state – the same request can be repeated after some time.
Response headers	X-IASReport-Signature	Base 64-encoded Report Signature.
	X-IASReport-Signing-Certificate	URL encoded Attestation Report Signing Certificate Chain in PEM format (all certificates in the chain, appended to each other).
	Request-ID	Random generated identifier for each request.

Response body	Attestation Verification Report serialized to a JSON string format:
	{
	"id":"<report_id>",
	"isvEnclaveQuoteStatus":"<quote_status>",
	"isvEnclaveQuoteBody":"<quote_body>",
	"platformInfoBlob":"<platform_info_blob><optional>",
	"revocationReason":<revocation_reason><optional>,
	"pseManifestStatus": "<pse_manifest_status><optional>",
	"pseManifestHash": "<pse_manifest_hash><optional>",
	"nonce":"<custom_value_passed_by_caller><optional>",
	"epidPseudonym":"<epid_pseudonym_for_linkable><optional>",
	"timestamp":"<timestamp>"
	}
	
	In case of an error during processing, the response body will be empty (an appropriate HTTP status code will define the problem and Request-ID header returned in case additional troubleshooting actions are required).	

3.2.3 Examples

3.2.3.1 Without PSE Manifest

HTTP request	
URI	POST https://test-as.sgx.trustedservices.intel.com:443/attestation/sgx/v2/report

Body	{ "isvEnclaveQuote": "AAEAAAEAAA+yth5<…*encoded_quote…*>GuOKBJ+5cs0PQcnZp" }	
HTTP response		
Status	200 OK	
Headers	Request-ID	de305d5475b4431badb2eb6b9e546014
	X-IASReport-Signature	lT6EiisC441buJNQhGZwl<…*signature…*>peqiMjar04nQR0AchJkw==
	X-IASReport-Signing-Certificate	-----BEGIN%20CERTIFICATE-----%0AMIIEoT<*...certificate_chain...*> GMnX%0A-----END%20CERTIFICATE-----%0A
Body	{ "id":"165171271757108173876306223827987629752", "isvEnclaveQuoteStatus":"OK", "isvEnclaveQuoteBody":"AAEAAAEAAA+yth5<…*encoded_quote_body…*>7h38CMfOng", "timestamp":" 2015-09-29T10:07:26.711023" }	

3.2.3.2 With PSE Manifest

HTTP request		
URI	POST https://test-as.sgx.trustedservices.intel.com:443/attestation/sgx/v2/report	
Body	{ "isvEnclaveQuote": "AAEAAAEAAA+yth5<…*encoded_quote…*>GuOKBJ+5cs0PQcnZp", "pseManifest": "AAAADsFbEHh9L4RmfOsLW<…*encoded_pse_manifest…*>2cKrl356PqfY3bh+A==" }	
HTTP response		
Status	200 OK	
Headers	Request-ID	de305d5475b4431badb2eb6b9e546014
	X-IASReport-Signature	lT6EiisC441buJNQhGZwl<…*signature…*>peqiMjar04nQR0AchJkw==
	X-IASReport-Signing-Certificate	-----BEGIN%20CERTIFICATE-----%0AMIIEoT<*...certificate_chain...*> GMnX%0A-----END%20CERTIFICATE-----%0A
Body	{ "id":"165171271757108173876306223827987629752", "isvEnclaveQuoteStatus":"OK", "isvEnclaveQuoteBody":"AAEAAAEAAA+yth5<…*encoded_quote_body…*>7h38CMfOng", "pseManifestStatus":"OK", "pseManifestHash":"DE75DD331267<…*encoded_pse_manifest_hash…*>4864716FF4B5", "timestamp":"2015-09-29T10:07:26.711023" }	

3.2.3.3 Quote with Linkable EPID Signature

HTTP request	
URI	POST https://test-as.sgx.trustedservices.intel.com:443/attestation/sgx/v2/report

Body	{ "isvEnclaveQuote": "AAEAAAEAAA+yth5<…*encoded_quote_with_linkable…*>J+5cs0PQcnZp" }	
HTTP response		
Status	200 OK	
Headers	Request-ID	de305d5475b4431badb2eb6b9e546014
	X-IASReport-Signature	lT6EiisC441buJNQhGZwl<…*signature…*>peqiMjar04nQR0AchJkw==
	X-IASReport-Signing-Certificate	-----BEGIN%20CERTIFICATE-----%0AMIIEoT<*...certificate_chain...*> GMnX%0A-----END%20CERTIFICATE-----%0A
Body	{ "id":"165171271757108173876306223827987629752", "isvEnclaveQuoteStatus":"OK", "isvEnclaveQuoteBody":"AAEAAAEAAA+yth5<…*encoded_quote_body…*>7h38CMfOng", "epidPseudonym":"2p4P9/<…*epid_pseudonym_structure*…>LbGUw8vUEPI/66x8ptZE=", "timestamp":" 2015-09-29T10:07:26.711023 " }	

3.2.3.4 With Invalid PSE Manifest

HTTP request		
URI	POST https://test-as.sgx.trustedservices.intel.com:443/attestation/sgx/v2/report	
Body	{ "isvEnclaveQuote": "AAEAAAEAAA+yth5<…*encoded_quote…*>GuOKBJ+5cs0PQcnZp", "pseManifest": "AAAADsFbEHh9L4RmfOsLW<…*encoded_invalid_pse_manifest…*>2cKrl356PqfY3bh+A==" }	
HTTP response		
Status	200 OK	
Headers	Request-ID	de305d5475b4431badb2eb6b9e546014
	X-IASReport-Signature	lT6EiisC441buJNQhGZwl<…*signature…*>peqiMjar04nQR0AchJkw==
	X-IASReport-Signing-Certificate	-----BEGIN%20CERTIFICATE-----%0AMIIEoT<*...certificate_chain...*> GMnX%0A-----END%20CERTIFICATE-----%0A
Body	{ "id":"59765165899944768216469568823557519409", "isvEnclaveQuoteStatus":"OK", "isvEnclaveQuoteBody":"AAEAAAEAAA+yth5<…*encoded_quote_body…*>7h38CMfOng", "pseManifestStatus":"INVALID", "pseManifestHash":"DE75DD331267<…*encoded_pse_manifest_hash…*>4864716FF4B5", "timestamp":"2015-09-29T10:13:48.279409" }	

3.2.3.5 With Nonce

HTTP request

URI	POST https://test-as.sgx.trustedservices.intel.com:443/attestation/sgx/v2/report	

Body	{
	"isvEnclaveQuote": "AAEAAAEAAAAAAAAADKB5Z<…*encoded_quote…*>AAAAAAAAAAAA==",
	"nonce": "0123456701234567"
	}	

HTTP response		
Status	200 OK	
Headers	Request-ID	de305d5475b4431badb2eb6b9e546014
	X-IASReport-Signature	lT6EiisC441buJNQhGZwl<…*signature…*>peqiMjar04nQR0AchJkw==
	X-IASReport-Signing-Certificate	-----BEGIN%20CERTIFICATE-----%0AMIIEoT<*...certificate_chain...*> GMnX%0A-----END%20CERTIFICATE-----%0A

Body	{
	"id":"94974578462868490675968868827087710 68",
	"isvEnclaveQuoteStatus":"OK",
	"isvEnclaveQuoteBody":"AAEAAAEAAA+yth5<…*encoded_quote_body…*>7h38CMfOng",
	"nonce":"0123456701234567",
	"timestamp":"2015-09-29T10:07:26.711023"
	}	

3.2.3.6 With Invalid Quote

HTTP request		
URI	POST https://test-as.sgx.trustedservices.intel.com:443/attestation/sgx/v2/report	

Body	{
	"isvEnclaveQuote": "AAAAADKB5Z<…*encoded_quote…*>AAAAAAAA=="
	}	

HTTP response		
Status	400 Bad Request	
Headers	Request-ID	de305d5475b4431badb2eb6b9e546014
Body	<empty>	

3.2.3.7 Revoked EPID Group

HTTP request		
URI	POST https://test-as.sgx.trustedservices.intel.com:443/attestation/sgx/v2/report	

Body	{
	“isvEnclaveQuote”: “AAAAADKB5Z<…*encoded_quote_for_revoked_group …*>AAAAAAAA==”
	}	

HTTP response		
Status	200 OK	

Headers	Request-ID	de305d5475b4431badb2eb6b9e546014
	X-IASReport-Signature	lT6EiisC441buJNQhGZwl<…*signature…*>peqiMjar04nQR0AchJkw==
	X-IASReport-Signing-Certificate	-----BEGIN%20CERTIFICATE-----%0AMIIEoT<*...certificate_chain...*> GMnX%0A-----END%20CERTIFICATE-----%0A

Body	{
	“id”:”6648460206045492248832007647790378406 3”,
	“isvEnclaveQuoteStatus”:”GROUP_REVOKED”,
	"isvEnclaveQuoteBody":"AAEAAAEAAA+yth5<…*encoded_quote_body…*>7h38CMfOng",
	“platformInfoBlob”:”150100650<…*pib_structure*…>7B094250DB00C610”,
	“revocationReason”:1,
	“timestamp”:”2015-09-29T10:07:26.711023“
	}	

4 Data Structures

The following chapter describes in detail the data structures used in the Attestation API.

4.1 Attestation Evidence Payload

Attestation Evidence Payload is a data structure submitted by the Service Provider to IAS so that identity of the ISV enclave and the validity of the platform can be verified.

Data format

Field name	Field type	Field value
isvEnclaveQuote	String	Base 64-encoded QUOTE structure generated by QE for the ISV enclave. See Quoting Data Structures for details. This field is ***mandatory***.
pseManifest	String	Base 64-encoded SGX Platform Service Security Property Descriptor structure provided by the platform. This field is ***optional***, it will be present only if ISV enclave uses SGX Platform Service.
nonce	String	A string that represents custom nonce value provided by SP. Maximum size of the nonce is 32 characters. This field is ***optional***, it is up to the SP to include that field. It can be used by SP to ensure that an old Attestation Verification Report cannot be reused in replay attacks. If this field is present, it will be returned back to SP as part of Attestation Verification Report.

4.2 Attestation Verification Report

The Attestation Verification Report is a data structure returned by the Attestation Service for Intel® SGX to the Service Provider. It contains a cryptographically signed report of verification of the identity of ISV enclave and the Trusted Computing Base (TCB) of the platform.

4.2.1 Report Data

Field name	Field type	Field value
id	Number	Integer that denotes a unique identifier of the Attestation Verification Report. This field is ***mandatory***.
timestamp	String	Representation of date and time the Attestation Verification Report was created. The time shall be in UTC and the encoding shall be compliant to ISO 8601 standard.

Field name	Field type	Field value
		This field is ***mandatory***.
isvEnclaveQuoteStatus	String	One of the following values: • **OK** – EPID signature of the ISV enclave QUOTE was verified correctly and the TCB level of the SGX platform is up-to-date. • **SIGNATURE_INVALID** – EPID signature of the ISV enclave QUOTE was invalid. The content of the QUOTE is not trustworthy. • **GROUP_REVOKED** – The EPID group has been revoked. When this value is returned, the revocationReason field of the Attestation Verification Report will contain revocation reason code for this EPID group as reported in the EPID Group CRL. The content of the QUOTE is not trustworthy. • **SIGNATURE_REVOKED** – The EPID private key used to sign the QUOTE has been revoked by signature. The content of the QUOTE is not trustworthy. • **KEY_REVOKED** – The EPID private key used to sign the QUOTE has been directly revoked (not by signature). The content of the QUOTE is not trustworthy. • **SIGRL_VERSION_MISMATCH** – SigRL version in ISV enclave QUOTE does not match the most recent version of the SigRL. In rare situations, after SP retrieved the SigRL from IAS and provided it to the platform, a newer version of the SigRL is made available. As a result, the Attestation Verification Report will indicate SIGRL_VERSION_MISMATCH. SP can retrieve the most recent version of SigRL from the IAS and request the platform to perform remote attestation again with the most recent version of SigRL. If the platform keeps failing to provide a valid QUOTE matching with the most recent version of the SigRL, the content of the QUOTE is not trustworthy. • **GROUP_OUT_OF_DATE** – The EPID signature of the ISV enclave QUOTE has been verified correctly, but the TCB level of SGX platform is outdated. The platform has not been identified as compromised and thus it is not revoked. It is up to the Service Provider to decide whether or not to trust the content of the QUOTE. This field is ***mandatory***.
isvEnclaveQuoteBody	String	Base 64-encoded BODY of QUOTE structure (i.e., QUOTE structure without signature related fields: SIG_LEN and SIG) as received in Attestation Evidence Payload. See Quoting Data Structures for details. This field is ***mandatory***.

Field name	Field type	Field value
revocationReason	Number	Integer corresponding to revocation reason code for a revoked EPID group listed in EPID Group CRL. Allowed values are described in RFC 5280. This field is ***optional***, it will only be present if value of isvEnclaveQuoteStatus is equal to GROUP_REVOKED.
pseManifestStatus	String	One of the following values: • **OK** – Security properties of the SGX Platform Service was verified as valid and up-to-date. • **UNKNOWN** – Security properties of the SGX Platform Service cannot be verified due to unrecognized PSE Manifest. • **INVALID** – Security properties of the SGX Platform Service are invalid. SP should assume the SGX Platform Service utilized by the ISV enclave is invalid. • **OUT_OF_DATE** – TCB level of SGX Platform Service is outdated but the Service has not been identified as compromised and thus it is not revoked. It is up to the SP to decide whether or not to assume the SGX Platform Service utilized by the ISV enclave is valid. • **REVOKED** – The hardware/firmware component involved in the SGX Platform Service has been revoked. SP should assume the SGX Platform Service utilized by the ISV enclave is invalid. • **RL_VERSION_MISMATCH** – A specific type of Revocation List used to verify the hardware/firmware component involved in the SGX Platform Service during the SGX Platform Service initialization process is out of date. If the SP rejects the remote attestation and forwards the Platform Info Blob to the SGX Platform SW through the ISV SGX Application, the SGX Platform SW will attempt to refresh the SGX Platform Service. This field is ***optional***, it will only be present if the SGX Platform Service Security Property Descriptor (pseManifest) is provided in Attestation Evidence Payload and isvEnclaveQuoteStatus is equal to OK or GROUP_OUT_OF_DATE.
pseManifestHash	String	SHA-256 calculated over SGX Platform Service Security Property Descriptor as received in Attestation Evidence Payload. This field is encoded using Base 16 encoding scheme. This field is ***optional***, it will only be present if pseManifest field is provided in Attestation Evidence Payload.
platformInfoBlob	String	A TLV containing an opaque binary blob that the Service Provider and the ISV SGX Application are supposed to forward to SGX Platform SW. This field is encoded using Base 16 encoding scheme.

Field name	Field type	Field value
		This field is ***optional***, it will only be present if one the following conditions is met: • isvEnclaveQuoteStatus is equal to GROUP_REVOKED or GROUP_OUT_OF_DATE, • pseManifestStatus is equal to one of the following values: OUT_OF_DATE, REVOKED or RL_VERSION_MISMATCH.
nonce	String	A string that represents a nonce value provided by SP in Attestation Evidence Payload. This field is ***optional***, it will only be present if nonce field is provided in Attestation Evidence Payload.
epidPseudonym	String	Byte array representing EPID Pseudonym that consists of the concatenation of EPID B (64 bytes) & EPID K (64 bytes) components of EPID signature. If two linkable EPID signatures for an EPID Group have the same EPID Pseudonym, the two signatures were generated using the same EPID private key. This field is encoded using Base 64 encoding scheme. This field is ***optional***, it will only be present if Attestation Evidence Payload contains Quote with *linkable* EPID signature.

4.2.2 Report Signature

The Attestation Verification Report is cryptographically signed by Report Signing Key (owned by the Attestation Service) using the RSA-SHA256 algorithm. The signature is calculated over the entire body of the HTTP response. Base 64-encoded signature is then returned in a custom HTTP response header X-IASReport-Signature.

In order to verify the signature over the report the following steps must be performed:

1. Decode and verify the Report Signing Certificate Chain that was sent together with the report (see Report Signing Certificate Chain for details). Make sure that the chain is rooted in a trusted Attestation Report Signing CA Certificate (available to download upon successful registration to IAS).
2. Optionally verify that the certificates in the chain have not been revoked (using CRLs indicated in the certificates).
3. Verify the signature over the report using Attestation Report Signing Certificate.

4.2.3 Report Signing Certificate Chain

The public part of Report Key is distributed in the form of an x.509 digital certificate called Attestation Report Signing Certificate. It is a leaf certificate issued by the Attestation Report Signing CA Certificate:

1) **Attestation Report Signing CA Certificate:** CN=Intel SGX Attestation Report Signing CA, O=Intel Corporation, L=Santa Clara, ST=CA, C=US
2) **Attestation Report Signing Certificate:** CN=Intel SGX Attestation Report Signing, O=Intel Corporation, L=Santa Clara, ST=CA, C=US

A PEM-encoded certificate chain consisting of Attestation Report Signing Certificate and Attestation Report Signing CA Certificate is returned in a custom HTTP response header X-IASReport-Signing-Certificate.

4.2.4 Platform Info Blob

Platform Info Blob TLV contains an opaque data structure to be forwarded from the Service Provider to the ISV SGX application. The ISV SGX application can then call the SGX SDK API sgx_report_attestation_status () for analysis. Internally, the *Platform Info Blob TLV* is a collection of status flags and platform TCB information wrapped in a TLV container (that includes a header). All *TLV header* ingredients are expressed in big-endian.

4.2.4.1 Platform Info Blob TLV

Name		Size (Bytes)	Description
TLV Header	Type	1	Identifier of Platform Info Blob TLV (*value: 21*).
	Version	1	Version of the data structure (*value: 1 or 2*).
	Size	2	The size of TLV Payload data that follows this field.
TLV Payload	Platform Info Blob	variable	Platform Information Blob to be processed by SGX Platform SW.

4.3 Quoting Data Structures

4.3.1 QUOTE Structure

Name		Offset (Bytes)	Size (Bytes)	Description
BODY	VERSION	0	2	Version of this structure. (Little-endian integer) • Value: 1 or 2
	SIGNATURE_TYPE	2	2	Type of the signature. Bit 0: 0 – unlinkable 1 – linkable Other bits reserved.
	GID	4	4	ID of platform's EPID Group. (Little-endian integer)
	ISVSVN_QE	8	2	The security version of the QE. (Little-endian integer)
	ISVSVN_PCE	10	2	The security version of the PCE. (Little-endian integer) This field is filled only in case of QUOTE with VERSION set to 2.

Name		Offset (Bytes)	Size (Bytes)	Description
				In case of QUOTE with VERSION set to 1, it is 0'ed.
	RESERVED	12	4	Reserved bytes (set to 0).
	BASENAME	16	32	EPID basename used in Quote.
REPORTBODY	CPUSVN	48	16	The security version of the CPU represented as a byte array.
	MISCSELECT	64	4	SSA frame extended feature set for the enclave. (Little-endian integer)
	RESERVED	68	28	Reserved bytes (set to 0).
	ATTRIBUTES	96	16	The values of the attributes flags for the enclave.
	MRENCLAVE	112	32	Enclave measurement represented as SHA256 digest (as defined in FIPS PUB 180-4).
	RESERVED	144	32	Reserved bytes (set to 0).
	MRSIGNER	176	32	SHA256 digest (as defined in FIPS PUB 180-4) of the big endian format modulus of the RSA public key of the enclave's signing key pair.
	RESERVED	208	96	Reserved bytes (set to 0).
	ISVPRODID	304	2	Enclave Product ID. (Little-endian integer)
	ISVSVN	306	2	The security version of the enclave. (Little-endian integer)
	RESERVED	308	60	Reserved bytes (set to 0).
	REPORTDATA	368	64	The value of REPORT.ReportData in REPORT input of GetQuote() or UserData in NB_UD input of GetQuote().
SIG_LEN		432	4	Length of SIG field in bytes. SIG_LEN is not part of the data the signature is based on. (Little-endian integer)
SIG		436	variable	Encrypted EPID signature over BODY and REPORTBODY.

4.4 SGX Platform Service Security Property Descriptor

SGX Platform Service Security Property Descriptor is an opaque 256 byte data structure provided by the platform.