IEEE P1363 / D13 (Draft Version 13). Standard Specifications for Public Key Cryptography

Annex B (Normative). Conformance.

Comments and suggestions are welcome. Please contact the chair, Ari Singer, at singerar@pb.com.

ANNEX B (Normative)
Conformance

The purpose of this Annex is to provide implementers with a consistent language for claiming conformance with parts of this standard. Note, however, that this Annex does not provide the means for verifying that a particular implementation indeed operates as claimed (this is sometimes called "implementation validation"). Therefore, conformance claims made by an implementation are mere claims, unless their accuracy can be assured by other means. Such other means may include, for example, implementation validation or assignment of legal liability to the implementer claiming conformance. They are outside the scope of this standard.

Note also that conformance for the purposes of this standard is a matter of functional correctness, not secure implementation; for the latter, implementers should refer to the security considerations in Annex D.

An implementation may claim conformance with one or more primitives, schemes or scheme operations specified in this standard, as further described in this Annex.

An implementation shall not claim conformance with this standard as a whole.

For background on primitives and schemes, please refer to Section 4. Specific primitives and schemes are defined in Sections 6–10.

B.1 General Model

A claim of conformance is an assertion by an implementation that it operates in accordance with some specification, over some set of inputs. Thus, a claim of conformance has fundamentally two parts:

— the specification with which conformance is claimed
— a set of inputs, or *conformance region*, for which the specification is defined, and over which conformance is claimed

For the purposes of this standard, the specification may be that of a primitive, a scheme operation, or a scheme. (An implementation may claim conformance with a scheme by claiming conformance with each operation in the scheme.) For a primitive, the inputs are those stated in the specification; for a scheme operation, the term "input" refers both to initial inputs such as messages, and inputs obtained during a step of the operation such as domain parameters, keys, and key derivation parameters. Recommended conformance regions are given in the specifications.

The set of inputs for which a specification is defined depends on the particular primitive or scheme. For a primitive, the set consists of all inputs that satisfy the assumptions stated for the primitive. For a scheme operation, the set includes at least those inputs that satisfy the assumptions for any primitives invoked by the operation. If the operation includes key validation or domain parameter validation, then the specification may also be defined for certain inputs that do not satisfy the assumptions for a primitive invoked by the scheme. Thus, for example, the specification of a scheme operation may be defined for invalid as well as valid keys when key validation is included in the scheme, even though the specification of a primitive invoked by the scheme is not defined for invalid keys. This is because the behavior of the scheme with key validation is defined as follows on invalid keys: the keys are rejected.

The minimum behavioral requirements for claiming conformance over a conformance region are as follows:

1. On all inputs in the conformance region, the implementation shall perform steps identical to or equivalent to those specified.

2. On all other inputs it accepts, the behavior of the implementation shall not interfere with correct operation on inputs in the conformance region. The behavior is otherwise unconstrained.

Acceptable behaviors in item 2 include operating in accordance with the specification (if the specification is defined for the input); rejecting the input; performing steps similar to those specified; or performing some other non-interfering operation.

Since primitives are intended for low-level software or hardware implementation, it may be inconvenient for an implementation of a primitive to check whether an input is supported. Consequently, while an implementation of a primitive may reject some unsupported inputs, it is not expected that an implementation of a primitive will reject every unsupported input. Primitives are not intended to provide security apart from schemes, so such checking is appropriately deferred to the schemes. It is expected that an implementation of a scheme will reject many or even all unsupported inputs, depending on whether key and domain parameter validation is included. For more discussion on the risks of not rejecting unsupported inputs, see Annex D.3.3.

An implementation may claim conformance over with more than one conformance region, or more than one specification.

NOTES

1—In the interest of interoperability, a conformance region should be sufficiently broad to support a range of possible applications. It is expected that implementation profiles for various applications will give minimum interoperability criteria, in terms of specifications and associated conformance region constraints. For a similar reason, a conformance region should be documented explicitly. (In some cases, however, the documentation may be implicit to some extent; for instance, the domain parameters may be unambiguously specified, but secret.)

2—Although an implementation's behavior is unconstrained on inputs outside the conformance region (except for not interfering with the behavior on inputs in the conformance region), it is recommended in the interest of robustness that an implementation include checks that prevent failure when specified assumptions are not satisfied. For instance, an implementation should include checks that prevent division by zero, or infinite loops, even if those checks are not necessary when the specified assumptions are satisfied.

3—The concept of "equivalence" (as in "perform steps ... equivalent to") should be understood in the sense of indistinguishability. A conformant implementation of a scheme or primitive may perform steps identical to those specified for the scheme or primitive, in the sense of performing those steps exactly as specified, or it may perform similar steps that produce the same observable behavior.

For instance, if a step calls for generating a random number, the implementation may generate a pseudorandom number. Under the usual cryptographic assumption that the pseudorandom generator is indistinguishable from a truly random generator, the implementation is equivalent to the specification at that step.

Similarly, an implementation may choose to apply restrictions that exclude certain rare events. For instance, an implementation may exclude DL or EC private keys that are equal to 1, and instead generate private keys in the range [2, r - 1]. An implementation with such a restriction will be indistinguishable from the specification, and may still claim conformance. On the other hand, an implementation that generates private keys in the range [1,1000] could not claim conformance, since its behavior would be observably different from the specification.

As another example, an implementation of IFES-RSA might output an error message when the output of the encryption primitive equals its input, which is a rare event.

B.2 Conformance Requirements

An implementation claiming conformance with a primitive or scheme specified in this standard shall meet the requirements specified in the sections of the standard indicated below, in addition to the general

criteria in B.1. Requirements are to be understood in the context of Sections 2 (References), 3 (Definitions and Acronyms), 4 (Types of Cryptographic Techniques) and 5 (Mathematical Conventions).

An implementation may claim conformance with a primitive, a scheme, or a scheme operation. For a scheme or scheme operation, conformance requirements for the selected primitive or primitives and for additional techniques such as encoding methods or key derivation functions are also assumed. When documenting conformance with a scheme, these scheme options shall be noted explicitly. In addition, the documentation shall indicate whether the implementation includes key validation or domain parameter validation, and, if so, what is validated—i.e., what properties of keys and parameters are assured by the validation. An implementation claiming conformance with a scheme shall satisfy the requirements for each operation in the scheme.

The following is a template for a claim of conformance:

"*Conforms with IEEE 1363-2000 (technique/options) over the region where (constraints on inputs).*"

The "technique/options" component identifies the primitive, scheme, or scheme operation; any underlying techniques such as the encoding method or hash function; and any additional choices such as whether and how domain parameter or key validation is performed. The "constraints on inputs" component identifies the conformance region. The method of expressing these components is left to the implementation. Some examples are given in B.3.

Primitive	Sections
DLSVDP-DH	4.2, 6.1, 6.2.1
DLSVDP-DHC	4.2, 6.1, 6.2.2
DLSVDP-MQV	4.2, 6.1, 6.2.3
DLSVDP-MQVC	4.2, 6.1, 6.2.4
DLSP-NR	4.2, 6.1, 6.2.5
DLVP-NR	4.2, 6.1, 6.2.6
DLSP-DSA	4.2, 6.1, 6.2.7
DLVP-DSA	4.2, 6.1, 6.2.8
ECSVDP-DH	4.2, 7.1, 7.2.1
ECSVDP-DHC	4.2, 7.1, 7.2.2
ECSVDP-MQV	4.2, 7.1, 7.2.3
ECSVDP-MQVC	4.2, 7.1, 7.2.4
ECSP-NR	4.2, 7.1, 7.2.5
ECVP-NR	4.2, 7.1, 7.2.6
ECSP-DSA	4.2, 7.1, 7.2.7
ECVP-DSA	4.2, 7.1, 7.2.8
IFEP-RSA	4.2, 8.1, 8.2.2
IFDP-RSA	4.2, 8.1, 8.2.1, 8.2.3
IFSP-RSA1	4.2, 8.1, 8.2.1, 8.2.4
IFVP-RSA1	4.2, 8.1, 8.2.5
IFSP-RSA2	4.2, 8.1, 8.2.1, 8.2.6
IFVP-RSA2	4.2, 8.1, 8.2.7
IFSP-RW	4.2, 8.1, 8.2.1, 8.2.8
IFVP-RW	4.2, 8.1, 8.2.9

Scheme	Operation	Sections

DL/ECKAS-DH1	key agreement	4.3, 9.1, 9.2
DL/ECKAS-DH2	key agreement	4.3, 9.1, 9.3
DL/ECKAS-MQV	key agreement	4.3, 9.1, 9.4
DL/ECSSA	signature generation	4.3, 10.1, 10.2.1, 10.2.2
	signature verification	4.3, 10.1, 10.2.1, 10.2.3
IFSSA	signature generation	4.3, 10.1, 10.3.1, 10.3.2
	signature verification	4.3, 10.1, 10.3.1, 10.3.3
IFES	encryption	4.3, 11.1, 11.2.1, 11.2.2
	decryption	4.3, 11.1, 11.2.1, 11.2.3

B.3 Examples

This section gives some examples of claims of conformance with the primitives and scheme operations in the standard.

B.3.1 DLSP-DSA

A hardware cryptographic module claims conformance with DLSP-DSA. The two parts of its conformance claim are as follows:

— *Specification:* DLSP-DSA, as given in Section 6.2.7
— *Conformance region:* Inputs of the form

  — the DL domain parameters q, r and g associated with the key s
  — the signer's private key s
  — the message representative, which is an integer $f \geq 0$

  where the DL domain parameters and the private key are valid and associated, subject to additional conditions that constrain the conformance region

An example of additional conditions is the following (this follows the recommendations in Section 6.2.7):

— the DL field order q is a 512-bit to 1024-bit prime
— the DL subgroup order r is a 160-bit prime
— the message representative f is at most 160 bits long

A module claiming conformance under these conditions may document its conformance as follows:

> *Conforms with IEEE 1363-2000 DLSP-DSA over the region where the DL field order q is a 512-bit to 1024-bit prime, the DL subgroup order r is a 160-bit prime, the domain parameters and the private key are valid and associated, and the message representative f is at most 160 bits long.*

Such a module may also claim conformance over a subset of the region just stated. For instance, it may claim conformance with the region specified in the Digital Signature Standard ([ANS97a] or [FIP94b]), where the DL field order q is a 512-bit, 576-bit, …, or 1024-bit prime and the subgroup order r and message representative f are as already stated.

B.3.2 DLSSA Signature Verification

A software application claims conformance with the DLSSA signature verification operation. The two parts of its conformance claim are:

— *Specification:* DLSSA signature verification, as given in Section 10.2.3, with a particular signature verification primitive and encoding method, and optionally with particular domain parameter and key validation methods
— *Conformance region:* Inputs of the form

  — the DL domain parameters q, r and g associated with the key w
  — the signer's purported public key w
  — the message M
  — the purported signature (c, d)

  subject to additional conditions that constrain the conformance region, some of which may depend on the specification.

An example of the particulars for DLSSA signature verification is the following:

— signature verification primitive: DLSP-DSA
— encoding method: EMSA1, with SHA-1 hash function
— no domain parameter or key validation

With this specification, the behavior is undefined for invalid domain parameters and keys. An implementation may thus claim conformance only over a conformance region consisting of valid domain parameters and keys. An example of the conditions that constrain the conformance region in this case is the following:

— domain parameters and public key are valid and associated
— the DL field order q is a 512-bit to 1024-bit prime
— the DL subgroup order r is a 160-bit prime
— message M is any that can be input to the implementation, at most 100 Mbytes long
— the purported signature (c, d) is any that can be input to the implementation, including at least all those such that c and d are in the range $[1, r - 1]$

A module claiming conformance under these conditions may document its conformance as follows (with shorthand for the specification):

> *Conforms with IEEE 1363-2000 DLSSA / DLVP-DSA / EMSA1 / SHA-1 signature verification operation with no explicit domain parameter or key validation over the region where the DL field order q is a 512-bit to 1024-bit prime, the DL subgroup order r is a 160-bit prime, the domain parameters and public key are valid and associated, the message M is at most 100 Mbytes long, and the purported signature (c, d) is any that can be input to the implementation, including at least all those such that c and d are in the range [1, r - 1].*

Another example of the particulars is:

— signature verification primitive: DLSP-DSA
— encoding method: EMSA1, with SHA-1 hash function
— "canonical seeded hash" domain parameter validation ([ANS97a]), and key validation (A.16.6)

With this specification, the behavior is defined for invalid domain parameters and public keys: they are rejected. (Indeed, domain parameters that are otherwise valid according to the definitions in this standard, but for which the seed is incorrect, are also rejected.) An implementation may thus claim conformance over a conformance region consisting of valid and invalid domain parameters and keys. The same example conditions as given above may be followed here, except for the condition that the domain parameters and public key are valid and associated.

An example conformance statement for this case is:

Conforms with IEEE 1363-2000 DLSSA / DLVP-DSA / EMSA1 / SHA-1 signature verification operation with "canonical seeded hash" domain parameter validation and key validation over the region where the DL field order q is a 512-bit to 1024-bit prime, the DL subgroup order r is a 160-bit prime, the message M is at most 100 Mbytes long, and the purported signature (c, d) is any that can be input to the implementation, including at least all those such that c and d are in the range [1, r – 1].

B.3.3 IFSP-RSA2

A hardware cryptographic module claims conformance with IFSP-RSA2. The two parts of its conformance claim are:

— *Specification:* IFSP-RSA2, as given in Section 8.2.6
— *Conformance region:* Inputs of the form

  — the signer's RSA private key K
  — the message representative, which is an integer f such that $0 \le f < n$,

  where the private key K is valid and such that $f \equiv 12 \pmod{16}$, subject to additional conditions that constrain the conformance region.

An example of additional conditions is the following (this follows the recommendations in Section 8.2.6):

— size of the modulus n in the private key is 512 to 2048 bits
— message representative f is in the range $[0, n - 1]$

A module claiming conformance under these conditions may document its conformance as follows:

Conforms with IEEE 1363-2000 IFSP-RSA2 over the region where the private key K is valid and the size of the modulus n is 512 to 2048 bits, and the message representative f º 12 (mod 16) is in the range [0, n – 1].

The module may also claim conformance over a subset of the region just stated. For instance, it may claim conformance with the region where the modulus size is 1024 to 2048 bits.

B.3.4 IFSSA Signature Verification

A software application claims conformance with the IFSSA signature verification operation. The two parts of its conformance claim are:

— *Specification:* IFSSA signature verification, as given in Section 10.3.3, with a particular signature verification primitive and encoding method, and optionally with a particular key validation method
— *Conformance region:* Inputs of the form

  — the signer's purported public key (n, e)
  — the message M
  — the purported signature s

  subject to additional conditions that constrain the conformance region, some of which may depend on the specification.

An example of the particulars for IFSSA signature verification is the following:

— signature verification primitive: IFVP-RSA2
— encoding method: EMSA2, with SHA-1 hash function
— no domain parameter or key validation

With this specification, the behavior is undefined for invalid keys. An implementation may thus claim conformance only over a conformance region consisting of valid keys. An example of the conditions that constrain the conformance region in this case is the following:

— public key (n, e) is valid
— size of the modulus n in the public key is 512 to 2048 bits
— message M is any that can be input to the implementation, at most 100 Mbytes long
— the purported signature s is any that can be input to the implementation, including at least all s in the range $[0, (n - 1)/2]$

The application may document its conformance as follows (with shorthand for the specification):

> *Conforms with IEEE 1363-2000 IFSSA / IFVP-RSA2 / EMSA2 / SHA-1 signature verification operation with no explicit key validation over the region where the public key (n, e) is valid, the size of the modulus n is 512 to 2048 bits, the message M is at most 100 Mbytes long, and the purported signature s is any that can be input to the implementation, including at least include all s in the range [0, (n – 1)/2].*