**SP 800-56A KAS Algorithm Request Form**

|  |  |
| --- | --- |
| **Company Name:** |  |
| **Implementation Name\*:** |  |
| **IGL Reviewer Name:** |  |
| **IGL Review Status:** | **as of** **<Date>** |

*\* The name of the Cryptographic Algorithm Implementation that contains this KAS (per Form-FIPS-50 – Algorithm Implementation and Information Form).*

**The NIST Key Agreement Schemes Validation System (KASVS) document is available at:** http://csrc.nist.gov/groups/STM/cavp/documents/keymgmt/KASVS.pdf

**Form Contents**

[General Information 2](#_Toc372543035)

[KAS Finite Field Cryptography (FFC) Information 3](#_Toc372543036)

[KAS Elliptic Curve Cryptography (ECC) Information 5](#_Toc372543037)

[Key Confirmation Information 7](#_Toc372543038)

[Prerequisite Testing Information 8](#_Toc372543039)

[KAS Assurances Questionnaire 9](#_Toc372543040)

*\*\*Please provide the information requested within each section of the form.\*\**

# General Information

*Select all Additional Functions performed (related to Assurances):*

1. Domain Parameter Generation

2. Domain Parameter Validation

3. Key Pair Generation

4. Full Validation (as specified in SP 800-56A Section 5.6.2.4 and/or 5.6.2.5)

5. Partial Validation (as specified in SP 800-56A Section 5.6.2.6, ECC only)

6. Key Regeneration

# KAS Finite Field Cryptography (FFC) Information

***Please select all that apply. (****Note:* *If more than one KAS FFC scheme is supported and parameter sets vary per scheme, multiple forms need to be completed to account for all test options.)*

|  |  |  |  |
| --- | --- | --- | --- |
| **Select Scheme:** | | | |
| **MQV1** | **MQV2** | **dhEphem** | **dhStatic** |
| **dhHybrid1** | **dhOneFlow** | **dhHybridOneFlow** |  |

|  |
| --- |
| **Select Key Derivation Function(s) (KDF(s)) implemented:** |
| **N/A: An SP 800-56A KDF is *not* implemented. (Only the DLC Primitive will be tested.)** |
| **Concatenation** |
| **ASN.1** |

|  |
| --- |
| **Select role(s) for Key Agreement:** |
| **Initiator** |
| **Responder** |

|  |
| --- |
| **(*dhStatic ONLY*) Select Nonce type(s) used in C(0,2) (Static):** |
| **1. Random Nonce** |
| **2. Time Stamp** |
| **3. Monotonically increasing sequence number** |
| **Combination of #2 and #3** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Select Parameter Set(s) and SHA used in KDF:**  *If an SP 800-56A KDF is not implemented, select* **one** *SHA used to hash the ZZ value before it is output.* | | | |
| **Parameter Set(s):** | **FB** | **FC** |
| *SHA Algorithm(s):* | SHA-224  SHA-256  SHA-384  SHA-512 | SHA-256  SHA-384  SHA-512 |

*Note: Length of Derived Secret Keying Material is automatically set to MAC Key Size.*

|  |  |  |
| --- | --- | --- |
| **If an SP 800-56A KDF is implemented, indicate one MAC (with attributes) supported for use in KAS testing:** | | |
| **CCM** | | |
| *Parameter Set:* | **FB** | **FC** |
| *Key Size:* | 128-bit  192-bit  256-bit | 128-bit  192-bit  256-bit |
| *Largest Nonce Length\*:* | 7  8  9  10  11  12  13 | 7  8  9  10  11  12  13 |
| *Largest MACLen (TagLen)\*:* | 8  10  12  14  16 | 8  10  12  14  16 |
| **CMAC** | | |
| *Parameter Set:* | **FB** | **FC** |
| *Key Size:* | 128-bit  192-bit  256-bit | 128-bit  192-bit  256-bit |
| *MACLen (TagLen)\*:* | 8 < = < = 16 | 8 < = < = 16 |
| **HMAC** | | |
| *Parameter Set:* | **FB** | **FC** |
| *Supporting SHA:* | SHA-224  SHA-256  SHA-384  SHA-512 | SHA-256  SHA-384  SHA-512 |
| *HMAC Key Size\*:* | (> = 14) | (> = 14) |
| *MACLen (TagLen)\*:* | (> = 8) | (> = 8) |

*\* Values in bytes.*

# KAS Elliptic Curve Cryptography (ECC) Information

***Please select all that apply. (****Note:* *If more than one KAS ECC scheme is supported and parameter sets vary per scheme, multiple forms need to be completed to account for all test options.)*

|  |  |  |  |
| --- | --- | --- | --- |
| **Select Scheme:** | | | |
| **Full Unified** | **Full MQV** | **Ephemeral Unified** | **Static Unified** |
| **One Pass DH** | **One Pass MQV** | **One Pass Unified** |  |

|  |
| --- |
| **Select Key Derivation Function(s) (KDF(s)) implemented:** |
| **N/A: An SP 800-56A KDF is *not* implemented. (Only the DLC Primitive will be tested.)** |
| **Concatenation** |
| **ASN.1** |

|  |
| --- |
| **Select role(s) for Key Agreement:** |
| **Initiator** |
| **Responder** |

|  |
| --- |
| **(*Static Unified ONLY*) Select Nonce type(s) used in C(0,2) (Static):** |
| **1. Random Nonce** |
| **2. Time Stamp** |
| **3. Monotonically increasing sequence number** |
| **Combination of #2 and #3** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Select Parameter Set(s), Curve, and SHA used in KDF:**  *Select* **one** *supported curve.*  *If an SP 800-56A KDF is not implemented, select* **one** *SHA used to hash the ZZ value before it is output.* | | | | |
| **Set:** | **EB** | **EC** | **ED** | **EE** |
| *Curve:* | P-224  K-233  B-233 | P-256  K-283  B-283 | P-384  K-409  B-409 | P-521  K-571  B-571 |
| *SHA:* | SHA-224  SHA-256  SHA-384  SHA-512 | SHA-256  SHA-384  SHA-512 | SHA-384  SHA-512 | SHA-512 |

*Note: Length of Derived Secret Keying Material is automatically set to MAC Key Size.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **If an SP 800-56A KDF is implemented, indicate one MAC (with attributes) supported for use in KAS testing:** | | | | |
| **CCM** | | | | |
| *Parameter Set:* | **EB** | **EC** |  |  |
| *Key Size:* | 128-bit  192-bit  256-bit | 128-bit  192-bit  256-bit |  |  |
| *Largest Nonce Len\*:* | 7  8  9  10  11  12  13 | 7  8  9  10  11  12  13 |  |  |
| *Largest MACLen (TagLen)\*:* | 14  16 | 16 |  |  |
| **CMAC** | | | | |
| *Parameter Set:* | **EB** | **EC** |  |  |
| *Key Size:* | 128-bit  192-bit  256-bit | 128-bit  192-bit  256-bit |  |  |
| *MACLen (TagLen)\*:* | 14 < = < = 16 | (must = 16) |  |  |
| **HMAC** | | | | |
| *Parameter Set:* | **EB** | **EC** | **ED** | **EE** |
| *Supporting SHA:* | SHA-224  SHA-256  SHA-384  SHA-512 | SHA-256  SHA-384  SHA-512 | SHA-384  SHA-512 | SHA-512 |
| *HMAC Key Size\*:* | (> = 14) | (> = 16) | (> = 24) | (> = 32) |
| *MACLen (TagLen)\*:* | (> = 14) | (> = 16) | (> = 24) | (> = 32) |

*\* Values in bytes.*

# Key Confirmation Information

***Please select all that apply, or select N/A.*** *(Note: If a KDF is not implemented, Key Confirmation is N/A.)*

**N/A – Key Confirmation Not Supported**

**Key Confirmation Supported**

**Nonce types used in Key Confirmation (see Section 5.4):**

(1) Random Nonce

(2) Time Stamp

(3) Monotonically increasing sequence number

(4) Combination of (2) and (3)

**Supported Key Confirmation roles:**

Provider

Recipient

**Supported Key Confirmation types:**

Unilateral

Bilateral (Provider & Recipient will be checked.)

**Notes:**

1. **dhEphem/Ephemeral Unified:** Key Confirmation is not possible (neither party has a static key pair).
2. **dhOneFlow/One Pass DH:** Provides Key Confirmation from V to U only. If Initiator is selected, Recipient will be tested. If Responder is selected, Provider will be tested.

Bilateral Key Confirmation is not provided.

# Prerequisite Testing Information

The supporting algorithms, as indicated below, must be tested and CAVP validated. Please provide the CAVP certificate number for each algorithm (if previously issued) ***or*** the name of the Cryptographic Algorithm Implementation that contains the algorithm (per Form-FIPS-50).

|  |  |  |  |
| --- | --- | --- | --- |
| **KAS FFC** | | | |
| DSA | *Required ONLY if any of the following Additional Functions is selected:*   * *Domain Parameter Generation 🡪 DSA PQG Gen & PQG Ver required.* * *Domain Parameter Validation 🡪 DSA PQG Ver required.* * *Key Pair Generation or Key Regeneration 🡪 DSA Key Pair Gen required.* | | |
| SHS | *Required* | | |
| RNG or DRBG | *Required* | | |
| CCM, CMAC, HMAC | *Required ONLY if Key Confirmation is supported (depending on MAC(s) selected)* | | |
|  | | | |
| **Prerequisite Algorithm** | **CAVP Cert. #** | ***or*** | **Implementation Name** |
| DSA |  | *or* |  |
| SHS |  | *or* |  |
| RNG |  | *or* |  |
| DRBG (per SP 800-90A) |  | *or* |  |
| CCM |  | *or* |  |
| CMAC |  | *or* |  |
| HMAC |  | *or* |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **KAS ECC** | | | |
| ECDSA | *Required ONLY if any of the following Additional Functions is selected:*   * *Full Validation or Key Regeneration 🡪 ECDSA PKV required.* * *Key Pair Generation or Key Regeneration 🡪 ECDSA Key Pair required.* | | |
| SHS | *Required* | | |
| RNG or DRBG | *Required* | | |
| CCM, CMAC, HMAC | *Required ONLY if Key Confirmation is supported (depending on MAC(s) selected)* | | |
|  | | | |
| **Prerequisite Algorithm** | **CAVP Cert. #** | ***or*** | **Implementation Name** |
| ECDSA |  | *or* |  |
| SHS |  | *or* |  |
| RNG |  | *or* |  |
| DRBG (per SP 800-90A) |  | *or* |  |
| CCM |  | *or* |  |
| CMAC |  | *or* |  |
| HMAC |  | *or* |  |

# KAS Assurances Questionnaire

Please check ALL applicable assurances implemented. At least one option must be selected from each group. (“N/A” may be selected for the assurances that do not apply to the implementation under test.)

This information will be used to determine SP 800-56A validation prerequisites, and which options to test.

|  |
| --- |
| **5.5.2 Assurances of Domain Parameter Validity** |
| Owner itself generates the set of domain parameters according to the requirements specified in §5.5.1. |
| Owner performs an explicit domain parameter validation as specified in:   * FIPS 186-3 for FFC based on parameter size selected in Table 1, **or** * ANSI X9.62-2 for ECC |
| Owner received assurance from a trusted third party (TTP) that the set of domain parameters was valid at the time they were generated by either 1 or 2 above. |
| **Additional information requested:** |
| Please provide a short summary of how this is implemented. |
| ⇨**Generate the parameters accordig to the requirements.** |
| If a TTP is used, who is the TTP? |
| ⇨ |
| Where is this implemented or where is the TTP implementation called (i.e., file name, function name, line no.)? |
| ⇨ |

|  |  |  |  |
| --- | --- | --- | --- |
| **5.6.2 Assurances of the Arithmetic Validity of a Public Key** | | | |
| ***5.6.2.1 OWNER Assurances of STATIC Public Key Validity*** | | | |
| N/A | Owner Full Validation | | TTP Full Validation |
| Owner Generation | TTP Generation | |  |
| ***5.6.2.2 RECIPIENT Assurances of STATIC Public Key Validity*** | | | |
| N/A | Recipient Full Validation | | TTP Full Validation |
| TTP Generation |  | |  |
| ***5.6.2.3 RECIPIENT Assurances of EPHEM Public Key Validity*** | | | |
| Claiming FIPS 140-2 IG 7.10: FFC or ECC Ephemeral scheme validation assurance not required. | | | |
| N/A | Recipient Full Validation | | TTP Full Validation |
| Recipient ECC Partial Validation (only for ECC) | | TTP ECC Partial Validation (only for ECC) | |
| **Additional information requested:** | | | |
| Please provide a short summary of how this is implemented. | | | |
| ⇨ | | | |
| Where is this implemented, or where is the TTP implementation called (i.e., file name, function name, line no.)? | | | |
| ⇨ | | | |

|  |  |  |
| --- | --- | --- |
| **5.6.3 Assurances Possession of Static Private Key** | | |
| ***5.6.3.1 OWNER Assurances of STATIC Private Key*** | | |
| N/A | | |
| OWNER receives assurance via: | Explicit Key Confirmation | Use of Encrypted Certificate |
| Key Regeneration | Trusted Provision | Key Generation |
| ***5.6.3.2 RECIPIENT Assurances of Owner’s Possession of a STATIC Private Key*** | | |
| N/A | 5.6.3.2.1 RECIPIENT obtains assurance through a TTP | |
| 5.6.3.2.2RECIPIENT obtains assurance directly from the Claimed Owner  Note: IUT must perform test with Key Agreement Scheme = RESPONDER and Key Confirmation role of PROVIDER for one of the following schemes: | | |
| 1. dhHybrid1Flow (One-Pass Unified) | 2. MQV1 (One-Pass MQV) | 3. dhOneFlow (One-Pass DH) |
| **Additional information requested:** | | |
| Please provide a short summary of how this is implemented. | | |
| ⇨ | | |
| Where is this implemented or where is the TTP implementation called (i.e., file name, function name, line no.)? | | |
| ⇨ | | |