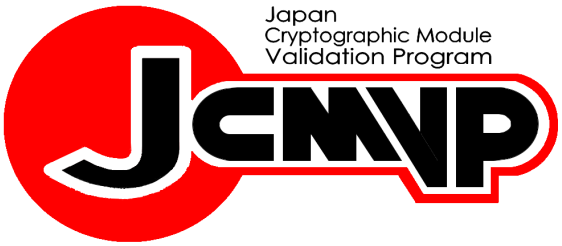


暗号アルゴリズム実装試験仕様書
– 鍵確立手法 –

令和元年 7 月 11 日

IPA

ATR-01-F

Cryptographic Algorithm Implementation Testing Requirements

独立行政法人情報処理推進機構

目 次

1 目的

本書は, 暗号アルゴリズム実装試験ツール (JCATT) に実装された鍵確立手法に関する暗号アルゴリズム実装試験仕様を記述する. 試験の対象とする暗号アルゴリズムは, 2 章に示す通りである.

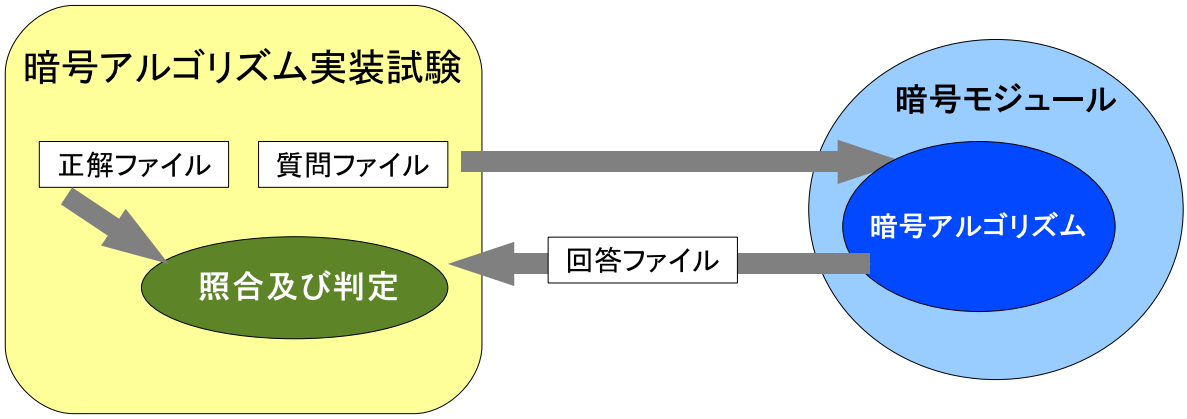
1.1 暗号アルゴリズム実装試験ツールの概要

暗号アルゴリズム実装試験ツールは次の特長を持つ.

- 試験対象の実装が暗号アルゴリズム仕様書に記述された事項に従って実装されているかどうかを試験する.
- 例えば鍵確立手法の場合は鍵共有, セッション鍵暗号化など, 各暗号が有する機能ごとに試験を行う.
- 暗号アルゴリズム実装試験ツールと試験対象の実装は, 各種ファイルを介してデータの通信を行う. このことにより, 様々なプラットフォーム上の暗号実装を試験可能となる. ここで, ツールで使う各種ファイルの内訳は以下のとおりである.
  - 質問ファイル: 暗号アルゴリズム実装試験ツールが生成するファイル. 暗号アルゴリズムに対する入力データ及び制御情報が記録されている. 暗号モジュール試験機関からベンダ側へ送る.
  - 正解ファイル: 暗号アルゴリズム実装試験ツールが質問ファイルと同時に生成するファイル. 暗号アルゴリズムに対する入力データ, 制御情報及び対応する出力データが記録されている. 暗号モジュール試験機関で保存し, 回答ファイルが送られてきた際に回答ファイルと照合する.
  - 回答ファイル: ベンダ側で, 質問ファイルを元に暗号モジュールが生成したテキストファイル. ベンダから暗号モジュール試験機関側へ送る.

ファイルフォーマットは文献 [14], サンプルファイルは文献 [15] を参照.
暗号アルゴリズム実装試験の流れは, 図 1.1 の通りである.

図 1.1: 暗号アルゴリズム実装試験の流れ



- 試験対象の実装が暗号アルゴリズム仕様書に記述された事項に従って実装されているかどうかを試験する.
- 例えば鍵確立手法の場合は鍵共有, セッション鍵暗号化など, 各暗号が有する機能ごとに試験を行う.
- 暗号アルゴリズム実装試験ツールと試験対象の実装は, 各種ファイルを介してデータの通信を行う. このことにより, 様々なプラットフォーム上の暗号実装を試験可能となる.

1.2 本書の構成

本書の以降の構成は次の通りである.

- 2 章: 暗号アルゴリズム実装試験ツールが試験の対象とする暗号アルゴリズムを示す.
- 3 章以降: 各暗号の試験項目を記述する.

なお, 本書を通して次の略語を使用する.

- JCATT: 暗号アルゴリズム実装試験ツール
- IUT: JCATT が試験の対象とする実装

2　本書で対象とするセキュリティ機能

本書が試験対象とする暗号アルゴリズムを次に示す.

2.1　承認されたセキュリティ機能

2.1.1　鍵確立手法

2.1.1.1　公開鍵確立手法

- Diffie-Hellman in NIST SP800-56A
  - dhHybrid1
  - dhEphem
  - dhHybridOneFlow
  - dhOneFlow
  - dhStatic
- Elliptic Curve Diffie-Hellman in NIST SP800-56A
  - (Cofactor) Full Unified Model
  - (Cofactor) Ephemeral Unified Model
  - (Cofactor) One-Pass Unified Model
  - (Cofactor) One-Pass Diffie-Hellman
  - (Cofactor) Static Unified Model
- Elliptic Curve Diffie-Hellman in SEC1
- Elliptic Curve Cofactor Diffie-Hellman in SEC1
- Key Establishment Schemes in NIST SP800-56B
  - Key-Agreement Schemes
    - KAS1
    - KAS2
  - Key-Transport Schemes
    - KTS-OAEP

2.1.1.2　鍵導出関数

- KDF in NIST SP800-108
  - KDF in Counter Mode
  - KDF in Feedback Mode
  - KDF in Double-Pipeline Iteration Mode
- PBKDF in NIST SP800-132
- KDF in NIST SP800-135
  - IKE version 1
  - IKE version 2
  - Key Derivation in TLS versions 1.0 and 1.1
  - Key Derivation in TLS version 1.2
  - Key Derivation Functions in ANS X9.42-2001 and ANS X9.63-2001
  - SSH Key Derivation Function
  - SRTP Key Derivation Function
  - SNMP Key Derivation Function

2.2 暗号アルゴリズム確認対象非承認セキュリティ機能

2.2.1 鍵確立手法

2.2.1.1 公開鍵確立手法

- PSEC-KEM

3 暗号アルゴリズム実装試験仕様 — 鍵確立手法 —

3.1 公開鍵確立手法

鍵共有アルゴリズム DH, ECDH , KAS1, KAS2, KTS-OAEP, KTS-KEM-KWS , PSEC-KEM の試験項目を記述する.

3.1.1 Diffie-Hellman (DH) in NIST SP800-56A

NIST SP800-56A [4] には, 次の 5 つの DH スキームが記述されている.

- dhHybrid1
- dhEphem
- dhHybridOneFlow
- dhOneFlow
- dhStatic

試験対象機能は, 各スキーム共に次の通りである.

- ドメインパラメータ生成機能
- ドメインパラメータ検証機能
- 鍵ペア生成機能
- 公開鍵検証機能
- 鍵共有機能

3.1.1.1 ドメインパラメータ生成機能試験

ドメインパラメータのうち p と q については,

- FIPS 186-4 Appendix A 1.1.2 “Generation of the Probable Primes p and q Using an Approved Hash Function”

又は

- FIPS 186-4 Appendix A 1.2.1 “Generation of the Primes p and q Using the Shawe-Taylor Algorithm”

に記述されているドメインパラメータ生成法に従って p と q が生成されていることを試験する.
また, ドメインパラメータ g については,

- FIPS 186-4 Appendix A 2.1 “Unverifiable Generation of the Generator g”

又は

- FIPS 186-4 Appendix A 2.3 “Verifiable Canonical Generation of the Generator g”

に記述されているドメインパラメータ生成法に従って g が生成されていることを試験する.
ドメインパラメータ生成機能は, 次の暗号アルゴリズムを組み合わせて使用する.

- FIPS 180-4 に記載されたハッシュ関数

ドメインパラメータ生成機能試験に先立って, この暗号アルゴリズムの暗号アルゴリズム実装試験に合格している必要がある.

3.1.1.1.1 p, q の生成試験

3.1.1.1.1.1 FIPS 186-4 A.1.1.2 に基づく確率的素数 p, q の生成

- 指定された FFC パラメータセット及びハッシュ関数に対して，IUT が生成した *domain_parameter_seed* 及び *counter* を JCATT に入力し, JCATT は FIPS 186-4 Appendix A 1.1.2 のアルゴリズムに従って 2 つの素数 p', q' を計算する. この p', q' と, IUT が生成した p, q がそれぞれ等しいこと.
- IUT が生成した複数 (別途規定する数) のドメインパラメータ (p,q) が全て異なるものであること.

3.1.1.1.1.2 FIPS 186-4 A.1.2.1 に基づく素数 p, q の生成

- 指定された FFC パラメータセット及びハッシュ関数に対して, IUT が生成した *firstseed* を JCATT に入力し，JCATT は FIPS 186-4 A.1.2.1 のアルゴリズムに従って p', q', *pseed*′, *qseed*′, *pgen_counter*′ 及び *qgen_counter*′ を計算する．この p', q', *pseed*′, *qseed*′, *pgen_counter*′ 及び *qgen_counter*′ と, IUT が生成した p, q, *pseed*, *qseed*, *pgen_counter* 及び *qgen_counter* とがそれぞれ等しいこと.
- IUT が生成した複数 (別途規定する数) のドメインパラメータ (p,q) が全て異なるものであること.

なお, ドメインパラメータ生成において使用するハッシュ関数は, FIPS 186-4 に従って次の通りとする. ($|p|$ を p のビット長, $|q|$ を q のビット長とする)

- $|p| = 2,048$, $|q| = 224 \cdots$ SHA-224 以上
- $|p| = 2,048$, $|q| = 256 \cdots$ SHA-256 以上
- $|p| = 3,072$, $|q| = 256 \cdots$ SHA-256 以上

3.1.1.1.2 g の生成試験

3.1.1.1.2.1 FIPS 186-4 A.2.1 に基づく g の生成

- p 及び q に対して, IUT が生成した g を JCATT に入力し, JCATT は FIPS 186-4 A.2.2 のアルゴリズムに従って $2 \leq g \leq p-1$ かつ $g^q \equiv 1 \mod p$ を満たすか試験する.

3.1.1.1.2.2 FIPS 186-4 A.2.3 に基づく g の生成

- p 及び q, *domain_parameter_seed*, 及び *index* に対して, IUT が生成した g を JCATT に入力し, JCATT は FIPS 186-4 A.2.3 のアルゴリズムに従って g' を計算する. この g' と IUT が生成した g とが等しいこと. (なお, p 及び q を FIPS 186-4 A.1.2.1 に基づいて生成する場合, *firstseed*, *pseed*, 及び *qseed* を連結したものを *domain_parameter_seed* として扱う.)

3.1.1.2 ドメインパラメータ検証機能試験

ドメインパラメータのうち p と q については,

- FIPS 186-4 Appendix A 1.1.3 “Validation of the Probable Primes p and q that were Generated Using an Approved Hash Function”

又は

- FIPS 186-4 Appendix A 1.2.2 “Validation of the DSA Primes p and q that were Constructed Using the Shawe-Taylor Algorithm”

に従って p と q を検証する機能を試験する.
また, ドメインパラメータ g については,

- FIPS 186-4 Appendix A 2.2 “Assurance of the Validity of the Generator g”

又は

- FIPS 186-4 Appendix A 2.4 “Validation Routine when the Canonical Generation of the Generator g Routine Was Used”

に従って g が検証する機能を試験する.
ドメインパラメータ検証機能は, 次の暗号アルゴリズムを組み合わせて使用する.

- FIPS 180-4 に記載されたハッシュ関数

ドメインパラメータ検証機能試験に先立って, この暗号アルゴリズムの暗号アルゴリズム実装試験に合格している必要がある.

3.1.1.2.1 p, q の検証試験

3.1.1.2.1.1 FIPS 186-4 A.1.1.3 に基づく p, q の検証

- 3.1.1.1.1.1 節に記述した p, q 生成機能に対する試験に適合するような p, q, *domain_parameter_seed*, *counter* に対して, IUT が “合格”と判定すること.
- 3.1.1.1.1.1 節に記述した p, q 生成機能に対する試験に違反するような p, q, *domain_parameter_seed*, *counter* に対して, IUT が “不正”と判定すること.

3.1.1.2.1.2 FIPS 186-4 A.1.2.2 に基づく p, q の検証

- 3.1.1.1.1.2 節に記述した p, q 生成機能に対する試験に適合するような p, q, $first\_seed$, $pseed$, $qseed$, $pgen\_counter$, $qgen\_counter$ に対して, IUT が “合格”と判定すること.
- 3.1.1.1.1.2 節に記述した p, q 生成機能に対する試験に違反するような p, q, $first\_seed$, $pseed$, $qseed$, $pgen\_counter$, $qgen\_counter$ に対して, IUT が “不正”と判定すること.

3.1.1.2.2 g の検証試験

3.1.1.2.2.1 FIPS 186-4 A.2.2 に基づく g の検証

- 3.1.1.1.2.1 節に記述した g 生成機能に対する試験に適合するような p, q, g に対して, IUT が “合格”と判定すること.
- 3.1.1.1.2.1 節に記述した g 生成機能に対する試験に違反するような p, q, g に対して, IUT が “不正”と判定すること.

3.1.1.2.2.2 FIPS 186-4 A.2.4 に基づく g の検証

- 3.1.1.1.2.2 節に記述した g 生成機能に対する試験に適合するような p, q, g, $domain\_parameter\_seed$, 及び $index$ に対して, IUT が “合格”と判定すること.
- 3.1.1.1.2.2 節に記述した g 生成機能に対する試験に違反するような p, q, g, $domain\_parameter\_seed$, 及び $index$ に対して, IUT が “不正”と判定すること.

3.1.1.3 鍵ペア生成機能試験

鍵ペア生成機能試験の試験項目は次の通りである.

- $y \equiv g^x \mod p$ であること
- $1 \leq x \leq q-1, 2 \leq y \leq p-2$ であること
- $y^q \equiv 1 \mod p$ であること
- IUT が生成した複数 (別途規定する数) の鍵ペアが全て異なるものであること.

鍵ペア生成機能は, 次の暗号アルゴリズムを組み合わせて使用する.

- NIST SP800-90A に記載された決定論的乱数生成器

鍵ペア生成機能試験に先立って, この暗号アルゴリズムの暗号アルゴリズム実装試験に合格している必要がある.

3.1.1.4 公開鍵検証機能試験

公開鍵検証機能試験の試験項目は次の通りである.

- 公開鍵 y 及びドメインパラメータ (p,q,g) が以下の条件全てを満たしている時には, “合格”と判定し, そうでなければ “不正”と判定すること.
  - $2 \leq y \leq p-2$ であること
  - $y^q \equiv 1 \mod p$ であること

3.1.1.5 鍵共有機能試験

鍵共有機能試験の試験項目は, 試験 1 又は試験 2 又は試験 3 である. 既定は試験 1 である.

Unilateral Key Confirmation をサポートする場合, 試験 1 に加えて, 試験 2 に合格する必要がある. NIST SP800-56A [4] の仕様上, dhEphem スキームを選択する場合, 試験 2 は選択できない.

Bilateral Key Confirmation をサポートする場合, 試験 1 に加えて, 試験 3 に合格する必要がある. NIST SP800-56A [4] の仕様上, dhEphem 及び dhOneFlow スキームを選択する場合, 試験 3 は選択できない.

試験 1, 試験 2 及び試験 3 で使用する KDF は, 表 4.2 に記載された KDF から指定する.

3.1.1.5.1 試験 1(既定の試験)

- JCATT が与えた *OtherInfo* の構成要素, 3.1.1.4 節に記述した公開鍵検証試験に適合するようなドメインパラメータ, 公開鍵 , HMAC を用いる鍵導出関数の選択時には *salt* に対して, IUT が正しい *DerivedKeyingMaterial*(DKM) 又は shared secret Z を生成すること.
- IUT が生成した複数 (別途規定する数) の共有鍵が全て異なるものであること.
- JCATT が与えたプライベート鍵, *OtherInfo* の構成要素, 3.1.1.4 節に記述した公開鍵検証試験に違反するような公開鍵に対して, IUT がエラーを出力すること.

3.1.1.5.2 試験 2(Unilateral Key Confirmation の試験)

3.1.1.5.2.1 Key Confirmation Provider の試験

- JCATT が与えた正しいドメインパラメータ, 公開鍵, プライベート鍵, *OtherInfo* の構成要素, *MacData*, 該当する場合には *Nonce* , HMAC を用いる鍵導出関数の選択時には *salt* に対して, IUT が正しい *KeyData* 及び *MacTag* を生成すること.

3.1.1.5.2.2 Key Confirmation Recipient の試験

- JCATT が与えた正しいドメインパラメータ, 公開鍵, プライベート鍵, *OtherInfo* の構成要素, *MacData*, *MacTag*, 該当する場合には *Nonce* , HMAC を用いる鍵導出関数の選択時には *salt* に対して, IUT が正しい *KeyData* を生成し, *MacTag* を検証合格と判定すること.
- JCATT が与えた改竄された公開鍵, プライベート鍵, *OtherInfo* の構成要素, *MacData*, *MacTag*, 該当する場合には *Nonce* , HMAC を用いる鍵導出関数の選択時には *salt* に対して, IUT がエラーを出力するか又は *MacTag* を検証不合格と判定すること.

3.1.1.5.3 試験 3(Bilateral Key Confirmation の試験)

- JCATT が与えた正しいドメインパラメータ, 公開鍵, プライベート鍵, *OtherInfo* の構成要素, Key Confirmation Provider の *MacTag*, 該当する場合には *Nonce* , HMAC を用いる鍵導出関数の選択時には *salt* に対して, IUT が正しい *KeyData* 及び Key Confirmation Recipient の *MacTag* を生成し, Key Confirmation Provider の *MacTag* を検証合格と判定すること.
- JCATT が与えた改竄された公開鍵, プライベート鍵, *OtherInfo* の構成要素, Key Confirmation Provider の *MacTag*, 該当する場合には *Nonce* , HMAC を用いる鍵導出関数の選択時には *salt* に対して, IUT がエラーを出力するか又は Key Confirmation Provider の *MacTag* を検証不合格と判定すること.

3.1.2 Elliptic Curve Diffie-Hellman (ECDH) in NIST SP800-56A

NIST SP800-56A [4] には, 次の 5 つの ECDH スキームが記載されている.

- (Cofactor) Full Unified Model
- (Cofactor) Ephemeral Unified Model
- (Cofactor) One-Pass Unified Model
- (Cofactor) One-Pass Diffie-Hellman
- (Cofactor) Static Unified Model

試験対象機能は, 各スキーム共に次の通りである.

- ドメインパラメータ生成機能
- ドメインパラメータ検証機能
- 鍵ペア生成機能
- 公開鍵検証機能
- 鍵共有機能

3.1.2.1 ドメインパラメータ生成機能試験

ドメインパラメータ生成機能試験の試験項目は次の通りである.

3.1.2.1.1 標数 p の場合

標数 p ドメインパラメータ生成機能試験の試験項目は次の通りである.

- セキュリティレベルを s として, n が $\max(2s, 224)$ ビット以上であること.
- $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$ であること.
- a, b, x_G, y_G が 0 以上 $p-1$ 以下の整数であること.
- ${y_G}^2 \equiv {x_G}^3 + ax_G + b \pmod{p}$ であること.
- n が素数であること.
- p が素数であること.
- $h \leq 2^{s/8}$ かつ $h = \lfloor (\sqrt{p}+1)^2/n \rfloor$ であること.
- $nG = \mathcal{O}$ であること.
- すべての $1 \leq B < 100$ に対して $p^B \not\equiv 1 \pmod{n}$ であること.
- $nh \neq p$ であること.
- IUT が生成した複数 (別途規定する数) のドメインパラメータが全て異なるものであること.

3.1.2.1.2 標数 2 の場合

標数 2 ドメインパラメータ生成機能試験の試験項目は次の通りである.

- セキュリティレベルを s として, n が $\max(2s, 224)$ ビット以上であること.
- $f(x)$ が次数 m のバイナリ既約多項式であること.
- a, b, x_G, y_G が次数 $m-1$ 以下のバイナリ多項式であること.
- $b \neq 0$ in $\mathbb{F}_{2^m}$ であること.
- ${y_G}^2 + x_G y_G \equiv {x_G}^3 + a{x_G}^2 + b$ in $\mathbb{F}_{2^m}$ であること.
- n が素数であること.
- $h \leq 2^{s/8}$ かつ $h = \lfloor (\sqrt{2^m}+1)^2/n \rfloor$ であること.
- $nG = \mathcal{O}$ であること.
- すべての $1 \leq B < 100$ に対して $2^{mB} \not\equiv 1 (\mod n)$ であること.
- $nh \neq 2^m$ であること.
- IUT が生成した複数 (別途規定する数) のドメインパラメータが全て異なるものであること.

3.1.2.2 ドメインパラメータ検証機能試験

ドメインパラメータ検証機能試験の試験項目は次の通りである.

- JCATT が与えた 3.1.2.1 節に記述したドメインパラメータ生成機能試験項目に適合するようなドメインパラメータに対して, IUT が “合格”と判定すること.
- JCATT が与えた 3.1.2.1 節に記述したドメインパラメータ生成機能試験項目に違反するようなドメインパラメータに対して, IUT が “不正”と判定すること.

3.1.2.3 鍵ペア生成機能試験

暗号アルゴリズム実装試験仕様書 — 公開鍵 — (ATR-01-A)[13] 3.1.2 節に記述した ECDSA の鍵ペア生成機能試験項目と同じである.

3.1.2.4 公開鍵検証機能試験

暗号アルゴリズム実装試験仕様書 — 公開鍵 — (ATR-01-A)[13] 3.1.2 節に記述した ECDSA の公開鍵検証機能試験項目と同じである.

3.1.2.5 鍵共有機能試験

鍵共有機能試験の試験項目は , 試験 1 又は試験 2 又は試験 3 である. 既定は試験 1 である.

Unilateral Key Confirmation をサポートする場合, 試験 1 に加えて, 試験 2 に合格する必要がある. NIST SP800-56A [4] の仕様上, (Cofactor) Ephemeral Unified Model スキームを選択する場合, 試験 2 は選択できない.

Bilateral Key Confirmation をサポートする場合, 試験 1 に加えて, 試験 3 に合格する必要がある. NIST SP800-56A [4] の仕様上, (Cofactor) Ephemeral Unified Model 及び (Cofactor) One-Pass Diffie-Hellman スキームを選択する場合, 試験 3 は選択できない.

試験 1, 試験 2 及び試験 3 で使用する KDF は, 表 4.6 に記載された KDF から指定する.

3.1.2.5.1 試験 1(既定の試験)

- JCATT が与えたプライベート鍵, *OtherInfo* の構成要素, 3.1.2.4 に記述した公開鍵検証試験に適合するようなドメインパラメータ, 公開鍵 , HMAC を用いる鍵導出関数の選択時には *salt* に対して, IUT が正しい *DerivedKeyingMaterial*(DKM) 又は shared secret *Z* を生成すること.
- IUT が生成した複数 (別途規定する数) の共有鍵が全て異なるものであること.
- JCATT が与えたプライベート鍵, *OtherInfo* の構成要素, 3.1.2.4 に記述した公開鍵検証試験に違反するようなドメインパラメータ, 公開鍵に対して, IUT がエラーを出力すること.

3.1.2.5.2 試験 2(Unilateral Key Confirmation の試験)

3.1.2.5.2.1 Key Confirmation Provider の試験

- JCATT が与えたプライベート鍵, *OtherInfo* の構成要素, 3.1.2.4 に記述した公開鍵検証試験に適合するようなドメインパラメータ, 公開鍵, *MacData*, 該当する場合には *Nonce* , HMAC を用いる鍵導出関数の選択時には *salt* に対して, IUT が正しい *KeyData* 及び *MacTag* を生成すること.

3.1.2.5.2.2 Key Confirmation Recipient の試験

- JCATT が与えたプライベート鍵, *OtherInfo* の構成要素, 3.1.2.4 に記述した公開鍵検証試験に適合するようなドメインパラメータ, 及び公開鍵, *MacData*, *MacTag*, 該当する場合には *Nonce* , HMAC を用いる鍵導出関数の選択時には *salt* に対して, IUT が正しい *KeyData* を生成し, *MacTag* を検証合格と判定すること.
- JCATT が与えた改竄された公開鍵, プライベート鍵, *OtherInfo* の構成要素, *MacData*, *MacTag*, 該当する場合には *Nonce* , HMAC を用いる鍵導出関数の選択時には *salt* に対して, IUT がエラーを出力するか又は *MacTag* を検証不合格と判定すること.

3.1.2.5.3 試験 3(Bilateral Key Confirmation の試験)

- JCATT が与えたプライベート鍵, *OtherInfo* の構成要素, 3.1.2.4 に記述した公開鍵検証試験に適合するようなドメインパラメータ, 及び公開鍵, Key Confirmation Provider の *MacTag*, 該当する場合には *Nonce*, HMAC を用いる鍵導出関数の選択時には *salt* に対して, IUT が正しい *KeyData* 及び Key Confirmation Recipient の *MacTag* を生成し, Key Confirmation Provider の *MacTag* を検証合格と判定すること.
- JCATT が与えた改竄された公開鍵, プライベート鍵, *OtherInfo* の構成要素, 3.1.2.4 に記述した公開鍵検証試験に適合するようなドメインパラメータ, Key Confirmation Provider の *MacTag*, 該当する場合には *Nonce*, HMAC を用いる鍵導出関数の選択時には *salt* に対して, IUT がエラーを出力するか又は Key Confirmation Provider の *MacTag* を検証不合格と判定すること.

3.1.3 Elliptic Curve (Cofactor) Diffie-Hellman (ECDH) in SEC1

ECDH in SEC1 の試験対象機能は次の通りである.

- ドメインパラメータ生成機能
- ドメインパラメータ検証機能
- 鍵ペア生成機能
- 公開鍵検証機能
- 鍵共有機能

3.1.3.1 ドメインパラメータ生成機能試験

暗号アルゴリズム実装試験仕様書 — 公開鍵 — (ATR-01-A)[13] 3.1.2 節に記述した ECDSA のドメインパラメータ生成機能試験項目と同じである.

3.1.3.2 ドメインパラメータ検証機能試験

暗号アルゴリズム実装試験仕様書 — 公開鍵 — (ATR-01-A)[13] 3.1.2 節に記述した ECDSA のドメインパラメータ検証機能試験項目と同じである.

3.1.3.3 鍵ペア生成機能試験

暗号アルゴリズム実装試験仕様書 — 公開鍵 — (ATR-01-A)[13] 3.1.2 節に記述した ECDSA の鍵ペア生成機能試験項目と同じである.

3.1.3.4 公開鍵検証機能試験

暗号アルゴリズム実装試験仕様書 — 公開鍵 — (ATR-01-A)[13] 3.1.2 節に記述した ECDSA の公開鍵検証機能試験項目と同じである.

3.1.3.5 鍵共有機能試験

鍵共有機能試験の試験項目は次の通りである.

- JCATT が与えた (複数の) プライベート鍵と公開鍵に対して, IUT が正しい共有鍵を生成すること.

KDF は, ANSI X9.63 [2] に記載された仕様の KDF を使用する. KDF で用いるハッシュ関数は, SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 の中から指定する.

3.1.4 KAS1 in NIST SP800-56B

KAS1 in NIST SP800-56B の試験対象機能は次の通りである.

- 鍵共有機能

3.1.4.1 鍵共有機能試験

KAS1 の鍵共有機能は, 次の暗号アルゴリズムを組み合わせて使用する.

- 表 4.11 に記載された鍵導出関数で使用されるハッシュ関数又は HMAC
- IUT が Party U を担う実装の場合, NIST SP 800-90A に記載された決定論的乱数生成器

鍵共有機能試験に先立って, これらの暗号アルゴリズム実装試験に合格している必要がある.

3.1.4.1.1 Party U の試験

KAS1-basic の Party U の試験の試験項目は次の試験 1 である. 既定は試験 1 である.
Key Confirmation をサポートする場合, 試験 1 に加えて試験 2 に合格する必要がある.

3.1.4.1.1.1 試験 1 (既定の試験. KAS1-basic の試験)

- 次のパラメータ及び指定された鍵導出関数に対して, IUT が正しい *DerivedKeyingMaterial*(DKM) を生成すること.
  - 公開鍵 (n, e),
  - DKM のビット長 $KBits$,
  - 鍵導出関数への入力ビット列 $OtherInfo$ の構成要素,
  - HMAC を用いる鍵導出関数の選択時,
    - HMAC の鍵として用いられる *salt*,

試験 1 で使用する鍵導出関数は, 表 4.11 から指定する.

3.1.4.1.1.2 試験 2 (Unilateral Key Confirmation の試験)

a. 表 3.1 の区分 I に $\surd$ の付いているパラメータ, 並びに指定された鍵導出関数及び Key Confirmation 用の指定された MAC アルゴリズムに対して, IUT が正しい *KeyData* を生成し $MacTag_V$ を検証合格と判定すること.

b. 表 3.1 の区分 II 又は III に $\surd$ の付いているパラメータ, 並びに指定された鍵導出関数及び Key Confirmation 用の指定された MAC アルゴリズムに対して, IUT がエラーを出力するか又は $MacTag_V$ を検証不合格と判定すること.

表 3.1: KAS1-Party_V-confirmation (for Party U) のパラメータ

No.	パラメータ	区分		
		I	II	III
1	公開鍵 (n,e)	√	√	√
2	secret value Z	√	√	√
3	暗号文 C	√	√	√
4	*DerivedKeyingMaterial*(DKM) のビット長 *KBits*	√	√	√
5	鍵導出関数への入力ビット列 $OtherInfo$ の構成要素	√		√
6	鍵導出関数への入力ビット列 $OtherInfo$ の改竄された構成要素		√	
7	HMAC を用いる鍵導出関数の選択時, HMAC の鍵として用いられる *salt*	√	√	√
8	*MacKey* のビット長	√	√	√
9	*MacData*	√	√	√
10	$MacTag_V$	√	√	
11	改竄された $MacTag_V$			√

3.1.4.1.2 Party V の試験

KAS1-basic の Party V の試験の試験項目は次の試験 1 である.
Key Confirmation をサポートする場合, 試験 1 に加えて試験 2 に合格する必要がある.

3.1.4.1.2.1 試験 1 (既定の試験. KAS1-basic の試験)

a. 表 3.2 の区分 I に $\surd$ の付いているパラメータ及び指定された鍵導出関数に対して, IUT が正しい *DerivedKeyingMaterial*(DKM) を生成すること.

b. 表 3.2 の区分 II に $\surd$ の付いているパラメータ及び指定された鍵導出関数に対して, IUT がエラーを出力し, DKM を生成しないこと.

表 3.2: KAS1-basic (for Party V) のパラメータ

No.	パラメータ	区分	
		I	II
1	プライベート鍵 (n,d)	$\surd$	$\surd$
2	暗号文 C	$\surd$	
3	RSASVE.Recover(...) がエラーを出力するような暗号文 C		$\surd$
4	DKM のビット長 $KBits$	$\surd$	$\surd$
5	鍵導出関数への入力ビット列 $OtherInfo$ の構成要素	$\surd$	$\surd$
6	HMAC を用いる鍵導出関数の選択時, HMAC の鍵として用いられる *salt*	$\surd$	$\surd$

試験 1 で使用する鍵導出関数は, 表 4.11 から指定する.

3.1.4.1.2.2 試験 2 (Unilateral Key Confirmation の試験)

a. 次のパラメータ, 並びに指定された鍵導出関数及び Key Confirmation 用の指定された MAC アルゴリズムに対して, IUT が正しい *KeyData* 及び $MacTag_V$ を生成すること.

- プライベート鍵 (n,d),
- 暗号文 C,
- *DerivedKeyingMaterial*(DKM) のビット長 $KBits$,
- 鍵導出関数への入力ビット列 $OtherInfo$ の構成要素,
- HMAC を用いる鍵導出関数の選択時,
  - HMAC の鍵として用いられる *salt*,
- *MacKey* のビット長,
- *MacData*,

3.1.5 KAS2 in NIST SP800-56B

KAS2 in NIST SP800-56B の試験対象機能は次の通りである.

- 鍵共有機能

3.1.5.1 鍵共有機能試験

KAS2 の鍵共有機能は, IUT が担う役割が Party U か Party V かによらず, 次の暗号アルゴリズムを組み合わせて使用する.

- 表 4.11 に記載された鍵導出関数で使用されるハッシュ関数又は HMAC
- NIST SP 800-90A に記載された決定論的乱数生成器

鍵共有機能試験に先立って, これらの暗号アルゴリズム実装試験に合格している必要がある.

3.1.5.1.1 Party U の試験

KAS2-basic の Party U の試験の試験項目は次の試験 1 である. Unilateral Key Confirmation をサポートする場合, 試験 1 に加えて試験 2 に合格する必要がある. Bilateral Key Confirmation をサポートする場合, 試験 1 に加えて試験 3 に合格する必要がある.

3.1.5.1.1.1 試験 1 (既定の試験. KAS2-basic の試験)

a. 表 3.3 の区分 I に $\surd$ の付いているパラメータ, 並びに指定された鍵導出関数に対して, IUT が正しい暗号文 C_U 及び *DerivedKeyingMaterial*(DKM) を生成すること.

b. 表 3.3 の区分 II に $\surd$ の付いているパラメータ, 並びに指定された鍵導出関数の一意の組みあわせに対して, 複数 (別途規定する数) 暗号文を生成させた時, 同じ暗号文 C_U が生成されないこと.

c. 表 3.3 の区分 III に $\surd$ の付いているパラメータ, 並びに指定された鍵導出関数に対して, IUT がエラーを出力し, DKM を生成しないこと.

表 3.3: KAS2-basic (for Party U) のパラメータ

No.	パラメータ	区分		
		I	II	III
1	Party U のプライベート鍵 (n_U, d_U)	$\surd$	$\surd$	$\surd$
2	Party V の公開鍵 (n_V, e_V)	$\surd$	$\surd$	$\surd$
3	Party V が生成した暗号文 C_V	$\surd$	$\surd$	
4	RSASVE.Recover(...) がエラーを出力するような暗号文 C_V			$\surd$
5	DKM のビット長 $KBits$	$\surd$	$\surd$	$\surd$
6	鍵導出関数への入力ビット列 $OtherInfo$ の構成要素	$\surd$	$\surd$	$\surd$
7	HMAC を用いる鍵導出関数の選択時, HMAC の鍵として用いられる *salt*	$\surd$	$\surd$	$\surd$

3.1.5.1.1.2 試験 2 (Unilateral Key Confirmation の試験)

Key Confirmation Provider (KAS2-Party_U-confirmation) の試験

a. 表 3.4 のパラメータ, 並びに指定された鍵導出関数及び Key Confirmation 用の指定された MAC アルゴリズムに対して, IUT が正しい暗号文 C_U, *KeyData* 及び $MacTag_U$ を生成すること.

表 3.4: KAS2-Party_U-confirmation (for Party U) のパラメータ

No.	パラメータ
1	Party U のプライベート鍵 (n_U, d_U)
2	Party V の公開鍵 (n_V, e_V)
3	Party V が生成した暗号文 C_V
4	*DerivedKeyingMaterial*(DKM) のビット長 $KBits$
5	鍵導出関数への入力ビット列 $OtherInfo$ の構成要素
6	HMAC を用いる鍵導出関数の選択時, HMAC の鍵として用いられる *salt*
7	*MacKey* のビット長

Key Confirmation Recipient (KAS2-Party_V-confirmation) の試験

a. 表 3.5 の区分 I に $\surd$ の付いているパラメータ, 並びに指定された鍵導出関数及び Key Confirmation 用の指定された MAC アルゴリズムに対して, IUT が正しい *KeyData* を生成し, $MacTag_V$ を検証合格と判定すること.

b. 表 3.5 の区分 II に $\surd$ の付いているパラメータ, 並びに指定された鍵導出関数及び Key Confirmation 用の指定された MAC アルゴリズムに対して, $MacTag_V$ を検証不合格と判定すること.

表 3.5: KAS2-Party_V-confirmation (for Party U) のパラメータ

No.	パラメータ	区分	
		I	II
1	Party U のプライベート鍵 (n_U, d_U)	$\surd$	$\surd$
2	Party U の secret value Z_U	$\surd$	$\surd$
3	Party U の暗号文 C_U	$\surd$	$\surd$
4	Party V の公開鍵 (n_V, e_V)	$\surd$	$\surd$
5	Party V が生成した暗号文 C_V	$\surd$	$\surd$
6	*DerivedKeyingMaterial*(DKM) のビット長 $KBits$	$\surd$	$\surd$
7	鍵導出関数への入力ビット列 $OtherInfo$ の構成要素	$\surd$	$\surd$
8	HMAC を用いる鍵導出関数の選択時, HMAC の鍵として用いられる *salt*	$\surd$	$\surd$
9	*MacKey* のビット長	$\surd$	$\surd$
10	$MacData_V$	$\surd$	$\surd$
11	$MacTag_V$	$\surd$	
12	改竄された $MacTag_V$		$\surd$

3.1.5.1.1.3 試験 3 (Bilateral Key Confirmation の試験)

a. 表 3.6 の区分 I に $\surd$ の付いているパラメータ, 並びに指定された鍵導出関数及び Key Confirmation 用の指定された MAC アルゴリズムに対して, IUT が正しい *KeyData* 及び $MacTag_U$ を生成し, $MacTag_V$ を検証合格と判定すること.

b. 表 3.6 の区分 II 又は III に $\surd$ の付いているパラメータ, 並びに指定された鍵導出関数及び Key Confirmation 用の指定された MAC アルゴリズムに対して, IUT がエラーを出力するか又は $MacTag_V$ を検証不合格と判定すること.

表 3.6: KAS2-bilateral-confirmation (for Party U) のパラメータ

No.	パラメータ	区分		
		I	II	III
1	Party U のプライベート鍵 (n_U, d_U)	√	√	√
2	Party U の secret value Z_U	√	√	√
3	Party U の暗号文 C_U	√	√	√
4	Party V の公開鍵 (n_V, e_V)	√	√	√
5	Party V が生成した暗号文 C_V	√	√	
6	改竄された暗号文 C_V			√
7	*DerivedKeyingMaterial*(DKM) のビット長 *KBits*	√	√	√
8	鍵導出関数への入力ビット列 $OtherInfo$ の構成要素	√	√	√
9	HMAC を用いる鍵導出関数の選択時, HMAC の鍵として用いられる *salt*	√	√	√
10	*MacKey* のビット長	√	√	√
11	$MacData_U$	√	√	√
12	$MacData_V$	√	√	√
13	$MacTag_V$	√		√
14	改竄された $MacTag_V$		√	

3.1.5.1.2 Party V の試験

KAS2-basic の Party V の試験の試験項目は次の試験 1 である. Unilateral Key Confirmation をサポートする場合, 試験 1 に加えて試験 2 に合格する必要がある. Bilateral Key Confirmation をサポートする場合, 試験 1 に加えて試験 3 に合格する必要がある.

3.1.5.1.2.1 試験 1 (既定の試験. KAS2-basic の試験)

a. 表 3.7 の対応項目 a に該当するパラメータ, 並びに指定された鍵導出関数に対して, IUT が正しい暗号文 C_V 及び *DerivedKeyingMaterial*(DKM) を生成すること.

b. 表 3.7 の対応項目 b に該当するパラメータ, 並びに指定された鍵導出関数の一意の組みあわせに対して, 複数 (別途規定する数) 暗号文を生成させた時, 同じ暗号文 C_V が生成されないこと.

c. 表 3.7 の対応項目 c に該当するパラメータ, 並びに指定された鍵導出関数に対して, IUT がエラーを出力し, DKM を生成しないこと.

表 3.7: KAS2-basic (for Party V) のパラメータ

No.	パラメータ	区分		
		I	II	III
1	Party U の公開鍵 (n_U, e_U)	√	√	√
2	Party U が生成した暗号文 C_U	√	√	
3	RSASVE.Recover(...) がエラーを出力するような暗号文 C_U			√
4	Party V のプライベート鍵 (n_V, d_V)	√	√	√
5	DKM のビット長 $KBits$	√	√	√
6	鍵導出関数への入力ビット列 $OtherInfo$ の構成要素	√	√	√
7	HMAC を用いる鍵導出関数の選択時, HMAC の鍵として用いられる *salt*	√	√	√

3.1.5.1.2.2 試験 2 (Unilateral Key Confirmation の試験)

Key Confirmation Recipient (KAS2-Party_U-confirmation) の試験

a. 表 3.8 の区分 I に $\surd$ の付いているパラメータ, 並びに指定された鍵導出関数及び Key Confirmation 用の指定された MAC アルゴリズムに対して, IUT が正しい *KeyData* を生成し, $MacTag_U$ を検証合格と判定すること.

b. 表 3.8 の区分 II に $\surd$ の付いているパラメータ, 並びに指定された鍵導出関数及び Key Confirmation 用の指定された MAC アルゴリズムに対して, IUT がエラーを出力するか又は $MacTag_U$ を検証不合格と判定すること.

表 3.8: KAS2-Party_U-confirmation (for Party V) のパラメータ

No.	パラメータ	区分	
		I	II
1	Party U の公開鍵 (n_U, e_U)	$\surd$	$\surd$
2	Party U が生成した暗号文 C_U	$\surd$	$\surd$
3	Party V のプライベート鍵 (n_V, d_V)	$\surd$	$\surd$
4	Party V の secret value Z_V	$\surd$	$\surd$
5	Party V の暗号文 C_V	$\surd$	$\surd$
6	*DerivedKeyingMaterial*(DKM) のビット長 *KBits*	$\surd$	$\surd$
7	鍵導出関数への入力ビット列 $OtherInfo$ の構成要素	$\surd$	$\surd$
8	HMAC を用いる鍵導出関数の選択時, HMAC の鍵として用いられる *salt*	$\surd$	$\surd$
9	*MacKey* のビット長	$\surd$	$\surd$
10	$MacData_U$	$\surd$	$\surd$
11	$MacTag_U$	$\surd$	
12	改竄された $MacTag_U$		$\surd$

Key Confirmation Provider (KAS2-Party_V-confirmation) の試験

a. 表 3.9 のパラメータ, 並びに指定された鍵導出関数及び Key Confirmation 用の指定された MAC アルゴリズムに対して, IUT が正しい C_V, *KeyData* 及び $MacTag_V$ を生成すること.

表 3.9: KAS2-Party_V-confirmation (for Party V) のパラメータ

No.	パラメータ
1	Party U の公開鍵 (n_U, e_U)
2	Party U が生成した暗号文 C_U
3	Party V のプライベート鍵 (n_V, d_V)
4	*DerivedKeyingMaterial*(DKM) のビット長 *KBits*
5	鍵導出関数への入力ビット列 $OtherInfo$ の構成要素
6	HMAC を用いる鍵導出関数の選択時, HMAC の鍵として用いられる *salt*
7	*MacKey* のビット長

3.1.5.1.2.3 試験 3 (Bilateral Key Confirmation の試験)

a. 表 3.10 の区分 I に $\surd$ の付いているパラメータ, 並びに指定された鍵導出関数及び Key Confirmation 用の指定された MAC アルゴリズムに対して, IUT が正しい *KeyData* 及び $MacTag_V$ を生成し, $MacTag_U$ を検証合格と判定すること.

b. 表 3.10 の区分 II 又は III に $\surd$ の付いているパラメータ並びに指定された鍵導出関数及び Key Confirmation 用の指定された MAC アルゴリズムに対して, IUT がエラーを出力するか又は $MacTag_U$ を検証不合格と判定すること.

表 3.10: KAS2-bilateral-confirmation (for Party V) のパラメータ

No.	パラメータ	区分		
		I	II	III
1	Party U の公開鍵 (n_U, e_U)	√	√	√
2	Party U が生成した暗号文 C_U	√	√	
3	改竄された暗号文 C_U			√
4	Party V のプライベート鍵 (n_V, d_V)	√	√	√
5	Party V の secret value Z_V	√	√	√
6	Party V が生成した暗号文 C_V	√	√	√
7	*DerivedKeyingMaterial*(DKM) のビット長 *KBits*	√	√	√
8	鍵導出関数への入力ビット列 $OtherInfo$ の構成要素	√	√	√
9	HMAC を用いる鍵導出関数の選択時, HMAC の鍵として用いられる *salt*	√	√	√
10	*MacKey* のビット長	√	√	√
11	$MacData_U$	√	√	√
12	$MacTag_U$	√		√
13	改竄された $MacTag_U$		√	
14	$MacData_V$	√	√	√

3.1.6 KTS-OAEP in NIST SP800-56B

KTS-OAEP in NIST SP800-56B の試験対象機能は次の通りである.

- 暗号化機能
- 復号機能

3.1.6.1 暗号化機能試験

暗号化機能試験の試験項目は次の試験 1, 又は試験 2 である. 既定は試験 1 である.
Unilateral Key Confirmation をサポートする場合, 試験 1 に加えて試験 3 に合格する必要がある.

3.1.6.1.1 試験 1 (既定の試験)

- 公開鍵 (n,e) 及び keying material K, 並びに指定されたハッシュ関数及びマスク生成関数 MGF 及び additional input A に対して IUT が生成した暗号文を, JCATT が復号した時に, もとの平文に復号されること.
- 同じ平文, 同じ公開鍵, 同じ additional input A の値に対して, 複数 (別途規定する数) 暗号文を生成させた時, 同じ暗号文が生成されないこと.

3.1.6.1.2 試験 2 (任意で実施する試験. 中間値 $mgfSeed$ を指定して行う既知入出力試験)

- 公開鍵 (n,e), keying material K 及び additional input A, 並びに指定されたハッシュ関数, マスク生成関数 MGF 及び中間値 $mgfSeed$ 対して正しい暗号文を IUT が生成すること.

3.1.6.1.3 試験 3 (Unilateral Key Confirmation の試験)

a. 表 3.11 の区分 I に $\surd$ の付いているパラメータ, 並びに指定されたハッシュ関数, マスク生成関数 MGF 及び Key Confirmation 用の指定された MAC アルゴリズムに対して, IUT が $MacTag_V$ を検証合格と判定すること.

b. 表 3.11 の区分 II に $\surd$ の付いているパラメータ, 並びに指定されたハッシュ関数, マスク生成関数 MGF 及び Key Confirmation 用の指定された MAC アルゴリズムに対して, IUT が $MacTag_V$ を検証不合格と判定すること.

表 3.11: KTS-OAEP-Party_V-confirmation (for Party U) のパラメータ

No.	パラメータ	区分	
		I	II
1	公開鍵 (n,e)	$\surd$	$\surd$
2	keying material $K(=MacKey\|\|KeyData)$	$\surd$	$\surd$
3	additional input A	$\surd$	$\surd$
4	中間値 $mgfSeed$	$\surd$	$\surd$
5	暗号文 C	$\surd$	$\surd$
6	$MacKey$ のビット長	$\surd$	$\surd$
7	$MacData_V$	$\surd$	$\surd$
8	$MacTag_V$	$\surd$	
9	改竄された $MacTag_V$		$\surd$

ハッシュ関数は, 表 4.17 の中から指定する.
マスク生成関数 MGF は, 表 4.18 の中から指定する.

KTS-OAEP の暗号化機能は, 次の暗号アルゴリズムを組み合わせて使用する.

- FIPS 180-4 又は FIPS 202 に記載されたハッシュ関数
- NIST SP 800-90A に記載された決定論的乱数生成器

暗号化機能試験に先立って, これらの暗号アルゴリズム実装試験に合格している必要がある.

3.1.6.2 復号機能試験

復号機能の試験項目は次の試験 1 である.
Unilateral Key Confirmation をサポートする場合, 試験 1 に加えて試験 2 に合格する必要がある.

3.1.6.2.1 試験 1 (既定の試験)

- 与えられたプライベート鍵 (n,d) と, 与えられた additional input A と, 指定されたハッシュ関数及びマスク生成関数 MGF と, 与えられた暗号文に対して, もとの keying material K に復号できること.
- 与えられたプライベート鍵 (n,d) と, 与えられた additional input A と, 指定されたハッシュ関数及びマスク生成関数 MGF と, 改竄された暗号文に対して不正検出を正しく行うこと.

3.1.6.2.2 試験 2 (Unilateral Key Confirmation の試験)

- 次のパラメータ, 並びに指定されたハッシュ関数, マスク生成関数 MGF, 及び Key Confirmation 用の指定された MAC アルゴリズムに対して, IUT が正しい $KeyData$ 及び $MacTag_V$ を生成すること.
  - プライベート鍵 (n,d),
  - additional input A,
  - 暗号文 C,
  - $MacKey$ のビット長,
  - $MacData_V$

ハッシュ関数は, 表 4.17 の中から指定する.
マスク生成関数 MGF は, 表 4.18 の中から指定する.

KTS-OAEP の復号機能は, 次の暗号アルゴリズムを組み合わせて使用する.

- FIPS 180-4 又は FIPS 202 に記載されたハッシュ関数

復号機能試験に先立って, これらの暗号アルゴリズム実装試験に合格している必要がある.

3.1.7 NIST SP800-56B の要素機能

次の要素機能に対する試験仕様を記述する.

- 鍵ペア生成機能
- 公開鍵部分検証機能
- 暗号化/復号プリミティブ
  - RSAEP
  - RSADP
- 暗号化/復号演算
  - RSASVE Generate 演算
  - RSASVE Recover 演算

3.1.7.1 鍵ペア生成機能試験

暗号アルゴリズム実装試験仕様書 — 公開鍵 — (ATR-01-A)[13] 3.1.4 節に記述した RSASSA-PKCS1-v1_5 の鍵ペア生成機能試験項目と同じである.

鍵ペア生成機能は, 次の暗号アルゴリズムを組み合わせて使用する.

- (FIPS 186-4 Appendix C.6 又は C.10 を用いる場合,) FIPS 180-4 に記載されたハッシュ関数
- NIST SP 800-90A に記載された決定論的乱数生成器

鍵ペア生成機能試験に先立って, これらの暗号アルゴリズム実装試験に合格している必要がある.

3.1.7.2 公開鍵部分検証機能試験

NIST SP 800-56B から参照される, NIST SP 800-89 [6] 5.3.3 “(Explicit) Partial Public Key Validation for RSA”に記載された検証機能を試験する.

- 公開鍵 (n,e) が以下の条件全てを満たしている時には, “合格”と判定し, そうでなければ “不正”と判定すること.
  - 法 n のビット長が 2048 又は 3072 のいずれかであること
  - $2^{16} < e < 2^{256}$ であること
  - n 及び e が奇数であること
  - n が合成数であり, n が素数のべき乗でないこと[a]
  - n が 751 以下の素因数をもたないこと

[a] FIPS 186-4 [7] に記載された Enhanced Miller-Rabin Test を用いる.

公開鍵部分検証機能は, 次の暗号アルゴリズムを組み合わせて使用する.

- NIST SP 800-90A に記載された決定論的乱数生成器 (FIPS 186-4 [7] に記載された Enhanced Miller-Rabin Test の中で参照される.)

公開鍵部分検証機能試験に先立って, これらの暗号アルゴリズム実装試験に合格している必要がある.

3.1.7.3 暗号化復号プリミティブ

3.1.7.3.1 RSAEP 要素機能試験

- 次のパラメータに対して正しい暗号文 c を IUT が生成すること.
  - 公開鍵 (n,e),
  - integer k,

3.1.7.3.2 RSADP 要素機能試験

1. プライベート鍵 (n,d), 暗号文 c に対して, IUT が正しい integer k を生成すること.
2. プライベート鍵 (n,d), RSADP(...) がエラーを出力するような暗号文 c に対して, IUT がエラーを出力すること.

3.1.7.4 暗号化/復号演算

3.1.7.4.1 RSASVE Generate 演算 要素機能試験

1. 与えられた公開鍵 (n,e) に対して IUT が生成した暗号文 C を, JCATT が復号した時に, IUT が生成したもとの secret value Z に復号されること.
2. 同じ公開鍵に対して, 複数 (別途規定する数) 暗号文を生成させた時, 同じ暗号文が生成されないこと.

RSASVE Generate 演算 要素機能は, 次の暗号アルゴリズムを組み合わせて使用する.

- NIST SP 800-90A に記載された決定論的乱数生成器

RSASVE Generate 演算 要素機能試験に先立って, これらの暗号アルゴリズム実装試験に合格している必要がある.

3.1.7.4.2 RSASVE Recover 演算 要素機能試験

1. プライベート鍵 (n,d), 暗号文 C に対して, IUT が正しい secret value Z を生成すること.
2. プライベート鍵 (n,d), RSASVE.Recover(...) がエラーを出力するような暗号文 C に対して, IUT がエラーを出力すること.

3.1.8 PSEC-KEM

PSEC-KEM の試験対象機能は次の通りである.

- 鍵ペア生成機能
- セッション鍵暗号化機能
- セッション鍵復号機能

それぞれの機能の試験項目を以下に記述する.

3.1.8.1 鍵ペア生成機能試験

暗号アルゴリズム実装試験仕様書 — 公開鍵 — (ATR-01-A)[13] 3.1.2 節に記述した ECDSA の鍵ペア生成機能試験項目と同じである.

3.1.8.2 セッション鍵暗号化機能試験

セッション鍵暗号化機能試験の試験項目は次の通りである.

- JCATT が与えた公開鍵 Q に対して, IUT が生成した暗号文を, JCATT で復号化した時に, 暗号文正当性の検証が合格となること.
- 同じ公開鍵に対して複数 (別途規定する数) の暗号文を生成させた時, IUT が同じ暗号文を生成しないこと.

KDF は, ISO/IEC 18033-2 [3] に記載された KDF1 から指定する. KDF1 で用いるハッシュ関数は, SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 の中から指定する.

3.1.8.3 セッション鍵復号機能試験

セッション鍵復号機能試験の試験項目は次の通りである.

- JCATT が与えた鍵ペア (d, Q) 及び暗号文に対して, IUT が正しく復号すること.
- JCATT が改竄した暗号文に対して, IUT が正しく “棄却”すること.

KDF は, ISO/IEC 18033-2 [3] に記載された KDF1 から指定する. KDF1 で用いるハッシュ関数は, SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 の中から指定する.

3.2 鍵導出関数

鍵導出関数の試験項目を記述する.

3.2.1 KDF in NIST SP 800-108

NIST SP800-108 [8] は, 次の 3 つの鍵導出関数を規定している.

- KDF in Counter Mode
- KDF in Feedback Mode
- KDF in Double-Pipeline Iteration Mode

3.2.1.1 鍵導出関数試験

NIST SP800-108 [8] で規定された KDF は, それを構成する要素機能である擬似ランダム関数 (PRF:Pseudorandom Function) に, FIPS 198-1 で規定された HMAC 又は NIST SP800-38B で規定された CMAC を用いる. ここで, PRF(HMAC 又は CMAC) の出力長 (ビット数) を h とする.

試験項目は次の通りである.

- IUT がサポートする, counter i の長さ (ビット数) r,
- JCATT が与えた
  - ランダムな鍵導出鍵 K_{I},
  - 導出する keying material (K_{O}) の長さ (ビット数) L,
  - IV (KDF in Feedback Mode 選択時のみ),

  及び
- ベンダーが指定する
  - $FixedInput := Label||0x00||Context$,
  - エンコードされた $[L]_2$

  に対して, IUT が正しい keying material (K_{O}) を生成すること.
- 導出する keying material の長さ L として, h で割り切れる値と, h で割り切れない値の両方を, IUT がサポートする範囲で試験すること.
- KDF in Feedback Mode 選択時, IV の長さを, 次の 4 つに分類し, IUT がサポートする範囲で試験すること. ($\mathbf{len}(IV)$ を IV のビット長とする)
  - $\mathbf{len}(IV) = 0$
  - $0 < \mathbf{len}(IV) < h$
  - $\mathbf{len}(IV) = h$
  - $\mathbf{len}(IV) > h$

ここで, NIST SP800-108 [8] では, $[L]_2$ の具体的なエンコーディングについて規定していない. そのため, ベンダが回答ファイルを通じて, エンコードされた $[L]_2$ を提供しなければならない. また, $FixedInput$ についても, 鍵導出関数の利用用途に応じて指定される値であることから, ベンダが回答ファイルを通じて, $FixedInput$ を提供しなければならない.

鍵導出関数は, 次の暗号アルゴリズムを組み合わせて使用する.

- FIPS 198-1 で規定された HMAC 又は NIST SP800-38B で規定された CMAC

鍵導出関数試験に先立って, この暗号アルゴリズムの暗号アルゴリズム実装試験に合格している必要がある.

3.2.2 PBKDF in NIST SP 800-132

3.2.2.1 鍵導出関数試験

NIST SP800-132 [9] で規定された Password-Based Key Derivation Function (PBKDF) は, それを構成する要素機能である擬似ランダム関数 (PRF:Pseudorandom Function) に, FIPS 198-1 で規定された HMAC を用いる. ここで, HMAC に組み合わせるハッシュ関数の出力長 (ビット数) を *hLen*, 入力ブロック長を B とする.

試験項目は次の通りである.

- JCATT が与えた
  - Password P,
  -
  - 導出する Master key (*mk*) の長さ (ビット数) *kLen*,
- IUT がサポートする
  - Iteration count C,

  及び
- IUT が生成する又は JCATT が与える
  - Salt S

  に対して, IUT が正しい Master key (*mk*) を生成すること.
- Password P の長さを, 次の 3 つに分類し, IUT がサポートする範囲で試験すること. ($\mathbf{len}(P)$ を P のビット長とする)
  - $112 \leq \mathbf{len}(P) < B$
  - $\mathbf{len}(P) = B$
  - $\mathbf{len}(P) > B$
- Salt S の長さを, 次の 3 つに分類し, IUT がサポートする範囲で試験すること. ($\mathbf{len}(S)$ を S のビット長とする)
  - $128 \leq \mathbf{len}(S) < (hLen - 32)$
  - $\mathbf{len}(S) = (hLen - 32)$
  - $\mathbf{len}(S) > (hLen - 32)$
- 導出する Master key の長さ *kLen* として, *hLen* で割り切れる値と, *hLen* で割り切れない値の両方を, IUT がサポートする範囲で試験すること.

鍵導出関数は, 次の暗号アルゴリズムを組み合わせて使用する.

- FIPS 198-1 で規定された HMAC

鍵導出関数試験に先立って, この暗号アルゴリズムの暗号アルゴリズム実装試験に合格している必要がある.

3.2.3 KDF in NIST SP 800-135

NIST SP800-135 [10] は, 次の 8 つの鍵導出関数を規定している.

- IKE version 1
- IKE version 2
- Key Derivation in TLS versions 1.0 and 1.1
- Key Derivation in TLS version 1.2

- Key Derivation Functions in ANS X9.42-2001 and ANS X9.63-2001
- SSH Key Derivation Function
- SRTP Key Derivation Function
- SNMP Key Derivation Function

3.2.3.1 IKE version 1 鍵導出関数試験

NIST SP800-135 [10] で規定された IKE version 1 の鍵導出関数群は, それを構成する要素機能である擬似ランダム関数 (PRF:Pseudorandom Function) に, HMAC を用いる.

鍵導出関数試験に先立って, HMAC の暗号アルゴリズム実装試験に合格している必要がある.

NIST SP800-135 [10] で規定された鍵導出関数群は, keying material *SKEYID*, *SKEYID_d*, *SKEYID_a*, *SKEYID_e* を導出する. IKE v1 の認証方式には次の 3 つの認証方式があり, keying material *SKEYID* の導出のやり方は, 選択された認証方式によって異なる.

- ディジタル署名を使う場合
- 公開鍵暗号を使う場合
- 事前共有鍵を使う場合

試験項目は次の通りである.

1. *SKEYID* の導出
   (a) 認証方式にディジタル署名を使う場合, JCATT が与えた次のパラメータに対して, IUT が正しい keying material *SKEYID* を導出すること.
      - ランダムな nonce *Ni_b*, *Nr_b*,
      - ランダムな shared secret g^{xy}.
   (b) 認証方式に公開鍵暗号を使う場合, JCATT が与えた次のパラメータに対して, IUT が正しい keying material *SKEYID* を導出すること.
      - ランダムな nonce *Ni_b*, *Nr_b*,
      - ランダムな cookie *CKY-I*, *CKY-R*.
   (c) 認証方式に事前共有鍵を使う場合, JCATT が与えた次のパラメータに対して, IUT が正しい keying material *SKEYID* を導出すること.
      - ランダムな nonce *Ni_b*, *Nr_b*,
      - ランダムな事前共有鍵 *pre-shared-key*.
2. *SKEYID_d*, *SKEYID_a*, *SKEYID_e* の導出
   (a) 既に導出された *SKEYID* 及び JCATT が与えた次のパラメータに対して, IUT が正しい keying material *SKEYID_d*, *SKEYID_a*, *SKEYID_e* を導出すること
      - ランダムな shared secret g^{xy},
      - ランダムな cookie *CKY-I*, *CKY-R*.

入力データの組の数は, 別途定める規定値とする.

3.2.3.2 IKE version 2 鍵導出関数試験

NIST SP800-135 [10] で規定された IKE version 2 の鍵導出関数群は, それを構成する要素機能である擬似ランダム関数 (PRF:Pseudorandom Function) に, HMAC を用いる. ここで, HMAC の出力長 (ビット数) を h とする.

鍵導出関数試験に先立って, HMAC の暗号アルゴリズム実装試験に合格している必要がある.

NIST SP800-135 [10] で規定された IKE version 2 の鍵導出関数群は, keying material *SKEYSEED*, *DKM*, Child SA 用の *DKM*, ephemeral Diffie-Hellman による shared secret g^{ir}(new) を用いた, Child SA 用の *DKM*, 及び IKE SA の “rekeying” のための *SKEYSEED* を導出する.

試験項目は次の通りである.

1. *SKEYSEED* の導出
   - JCATT が与えた次のパラメータに対して, IUT が正しい keying material *SKEYSEED* を導出すること.
     - ランダムな nonce *Ni*, *Nr*,
     - ランダムな shared secret g^{ir}.
2. *DKM* の導出
   - 既に導出された *SKEYSEED* 及び JCATT が与えた次のパラメータに対して, IUT が正しい keying material *DKM* を導出すること
     - ランダムな nonce *Ni*, *Nr*,
     - ランダムな Security Parameter Index *SPIi*, *SPIr*,
3. Child SA 用 *DKM* の導出
   - 既に導出された *SKEYSEED* の先頭 *h* ビットで表現される *SK_d* 及び JCATT が与えた次のパラメータに対して, IUT が正しい keying material *DKM* を導出すること
     - ランダムな nonce *Ni*, *Nr*.
4. ephemeral Diffie-Hellman による shared secret g^{ir}(new) を用いた, Child SA 用 *DKM* の導出
   - 既に導出された *SKEYSEED* の先頭 *h* ビットで表現される *SK_d* 及び JCATT が与えた次のパラメータに対して, IUT が正しい keying material *DKM* を導出すること
     - ランダムな nonce *Ni*, *Nr*,
     - ランダムな shared secret g^{ir}(new).
5. *SKEYSEED* (rekeying) の導出
   - 既に導出された *SKEYSEED* の先頭 *h* ビットで表現される *SK_d* 及び JCATT が与えた次のパラメータに対して, IUT が正しい keying material *SKEYSEED* を導出すること.
     - ランダムな nonce *Ni*, *Nr*,
     - ランダムな shared secret g^{ir}(new).

入力データの組の数は, 別途定める規定値とする.

3.2.3.3 Key Derivation in TLS versions 1.0 and 1.1 鍵導出関数試験

TLS version 1.0 及び version 1.1 の鍵導出関数は, それを構成する要素機能である擬似ランダム関数 (PRF:Pseudorandom Function) を, HMAC-MD5 及び HMAC-SHA-1 を組みあわせて構成する.

鍵導出関数試験に先立って, HMAC-SHA-1 の暗号アルゴリズム実装試験に合格している必要がある.

TLS version 1.1 の鍵導出関数は, まず, *pre_master_secret*, *ServerHello.Random*, 及び *ClientHello.Random* を与え, keying material *master_secret* を導出する. TLS version 1.2 の鍵導出関数は, 次に, 導出された *master_secret*, *server_random*, 及び *client_random* を与え, keying material *key_block* を導出する.

TLS version 1.1 では, 鍵確立に RSA を用いる場合と Diffie-Hellman を用いる場合とで, 鍵導出関数の入力となる *pre_master_secret* の取扱いに差がある. 前者は, 48 バイトの固定長の *pre_master_secret* を取り扱うのに対し, 後者は, 具体的には, Diffie-Hellman を用いて shared secret (*Z*) を確立し, バイト列に変換した *Z* の Most Significant Byte (MSB) が 0x00 の場合は, MSB が 0x00 で無くなるまで先頭の 0x00 を取り除いたものを *pre_master_secret* として取り扱う.

試験項目は次の試験 1 又は試験 2 である. 既定は試験 1 である.

試験 1(既定の試験)

1. JCATT が与えた次のパラメータに対して, IUT が正しい keying material *master_secret* を導出すること.
   - ランダムな *pre_master_secret*,
   - ランダムな *ServerHello.Random*, *ClientHello.Random*
2. 既に導出された *master_secret* 及び JCATT が与えた次のパラメータに対して, IUT が正しい keying material *key_block* を導出すること.
   - ランダムな *server_random*, *client_random*.

入力データの組の数は, 別途定める規定値とする.

試験 2(Diffie-Hellman を用いて *pre_master_secret* を確立する場合の試験)

1. JCATT が与えた次のパラメータに対して, IUT が正しい keying material *master_secret* を導出すること.
   - ランダムな *Z* (先頭数バイトが 0x00 になっている場合を含む),
   - ランダムな *ServerHello.Random*, *ClientHello.Random*
2. 既に導出された *master_secret* 及び JCATT が与えた次のパラメータに対して, IUT が正しい keying material *key_block* を導出すること.
   - ランダムな *server_random*, *client_random*

入力データの組の数は, 別途定める規定値とする.

3.2.3.4 Key Derivation in TLS version 1.2 鍵導出関数試験

TLS version 1.2 の鍵導出関数は, それを構成する要素機能である擬似ランダム関数 (PRF:Pseudorandom Function) に, HMAC を用いる.
鍵導出関数試験に先立って, HMAC の暗号アルゴリズム実装試験に合格している必要がある.

TLS version 1.2 の鍵導出関数は, まず, *pre_master_secret*, *ServerHello.Random*, 及び *ClientHello.Random* を与え, keying material*master_secret* を導出する. TLS version 1.2 の鍵導出関数は, 次に, 導出された *master_secret*, *server_random*, 及び *client_random* を与え, keying material *key_block* を導出する.

TLS version 1.2 では, 鍵確立に RSA を用いる場合と Diffie-Hellman を用いる場合とで, 鍵導出関数の入力となる *pre_master_secret* の取扱いに差がある. 前者は, 48 バイトの固定長の *pre_master_secret* を取り扱うのに対し, 後者は, 具体的には, Diffie-Hellman を用いて shared secret (*Z*) を確立し, バイト列に変換した *Z* の Most Significant Byte (MSB) が 0x00 の場合は, MSB が 0x00 で無くなるまで先頭の 0x00 を取り除いたものを *pre_master_secret* として取り扱う.

試験項目は次の試験 1 又は試験 2 である. 既定は試験 1 である.

試験 1(既定の試験)

1. JCATT が与えた次のパラメータ及び指定された PRF に対して, IUT が正しい keying material *master_secret* を導出すること.
   - ランダムな *pre_master_secret*,
   - ランダムな *ServerHello.Random*, *ClientHello.Random*
2. 既に導出された *master_secret*, JCATT が与えた次のパラメータ, 及び及び指定された PRF に対して, IUT が正しい keying material *key_block* を導出すること.
   - ランダムな *server_random*, *client_random*.

入力データの組の数は, 別途定める規定値とする.

試験 2(Diffie-Hellman を用いて *pre_master_secret* を確立する場合の試験)

1. JCATT が与えた次のパラメータ及び指定された PRF に対して, IUT が正しい keying material *master_secret* を導出すること.
   - ランダムな *Z* (先頭数バイトが 0x00 になっている場合を含む),
   - ランダムな *ServerHello.Random*, *ClientHello.Random*
2. 既に導出された *master_secret*, JCATT が与えた次のパラメータ, 及び指定された PRF に対して, IUT が正しい keying material *key_block* を導出すること.
   - ランダムな *server_random*, *client_random*

入力データの組の数は, 別途定める規定値とする.

3.2.3.5 Key Derivation Functions in ANS X9.42-2001 and ANS X9.63-2001 鍵導出関数試験

ANS X9.42-2001(又は ANS X9.63-2001) の鍵導出関数は, ハッシュ関数を直接用いる.

鍵導出関数試験に先立って, ハッシュ関数の暗号アルゴリズム実装試験に合格している必要がある.

試験項目は次の通りである.

1. JCATT が与えた次のパラメータに対して, IUT が正しい keying material *KeyData* を導出すること.
   - ランダムな shared secret *Z*,
   - ランダムな *OtherInfo*(ANS X9.63-2001 では *SharedInfo*).

入力データの組の数は, 別途定める規定値とする.

なお, ANS X9.42-2001 の鍵導出関数は, ANS X9.42-2003[1] の鍵導出関数と同一である. また, ANS X9.63-2001 の鍵導出関数は, ANS X9.63-2011 の鍵導出関数と同一である.

3.2.3.6 SSH 鍵導出関数試験

SSH の鍵導出関数は, ハッシュ関数を直接用いる. 鍵導出関数試験に先立って, ハッシュ関数の暗号アルゴリズム実装試験に合格している必要がある.

SSH の鍵導出関数は shared secret K, H, 及び *session_id* を与え, keying material *Initial IV*(*client to server*), *Initial IV*(*server to client*), *Encryption key*(*client to server*), *Encryption key*(*server to client*), *Integrity key*(*client to serv*
Integrity key(*server to client*) を導出する. この時, shared secret K を RFC 4251 で定義された mpint の表現にエンコードして, ハッシュ関数に入力する必要がある. NIST ASKDFVS[11] では mpint の表現にエンコードされた shared secret K を質問ファイルで指定している.

試験項目は次の通りである.

1. JCATT が与えた次のパラメータ:
   - ランダムな shared secret K,
   - ランダムな H,
   - ランダムな *session_id*

   に対して, IUT が正しい
   - *Initial IV*(*client to server*),
   - *Initial IV*(*server to client*),
   - *Encryption key*(*client to server*),
   - *Encryption key*(*server to client*),
   - *Integrity key*(*client to server*),
   - *Integrity key*(*server to client*)

   を導出すること.

入力データの組の数は, 別途定める規定値とする.

3.2.3.7 SRTP 鍵導出関数試験

SRTP の鍵導出関数は, ブロック暗号 AES を直接用いる. 鍵導出関数試験に先立って, AES の暗号アルゴリズム実装試験に合格している必要がある.

SRTP の鍵導出関数は master key (*k_master*), master salt (*master_salt*), key-derivation rate (*kdr*), SRTP 用の *index*, SRTCP 用の *index* を与え, keying material SRTP 用セッション暗号鍵 (*k_e*), SRTP 用セッションメッセージ認証鍵 (*k_a*), SRTP 用セッションソルト鍵 (*k_s*), SRTCP 用セッション暗号鍵 (*k_e*), SRTCP 用セッションメッセージ認証鍵 (*k_a*), SRTCP 用セッションソルト鍵 (*k_s*) を導出する.

試験項目は次の通りである.

1. JCATT が与えた次のパラメータ:
   - ランダムな master key *k_master*,
   - ランダムな master salt *master_salt*,
   - key-derivation rate *kdr*,
   - SRTP 用の *index*,
   - SRTCP 用の *index*

   に対して, IUT が正しい
   - SRTP 用セッション暗号鍵 *k_e*,
   - SRTP 用セッションメッセージ認証鍵 *k_a*,
   - SRTP 用セッションソルト鍵 *k_s*,
   - SRTCP 用セッション暗号鍵 *k_e*,
   - SRTCP 用セッションメッセージ認証鍵 *k_a*,
   - SRTCP 用セッションソルト鍵 *k_s*

   を導出すること.

入力データの組の数は, 別途定める規定値とする.

3.2.3.8 SNMP 鍵導出関数試験

SNMP の鍵導出関数は, ハッシュ関数を直接用いる. 鍵導出関数試験に先立って, ハッシュ関数の暗号アルゴリズム実装試験に合格している必要がある.

SNMP の鍵導出関数は *password*, 及び *snmpEngineID* を与え, keying material *key* を導出する.

試験項目は次の通りである.

1. JCATT が与えた次のパラメータ:
   - ランダムな *password*,
   - ランダムな *snmpEngineID*

   に対して, IUT が正しい *key* を導出すること.

入力データの組の数は, 別途定める規定値とする.

4 確認書発行条件

4.1 パラメータについて

鍵確立手法において, 暗号アルゴリズム確認書を発行するための条件は, 網掛けされた試験対象機能を少なくとも 1 個実装し, 暗号アルゴリズム実装試験に合格することである. 暗号アルゴリズム実装試験に使用するパラメータの入力条件及びその既定値は, 表 4.1～ 表 4.33 に記載する値とする.

4.1.1 公開鍵確立手法

4.1.1.1 DH in NIST SP800-56A

表 4.1: DH in NIST SP800-56A の既定値及び入力条件

試験対象機能	入力欄		既定値	入力条件
ドメインパラメータ生成	FFC パラメータセット		FC	FB, FC のいずれか
	試験の選択		FIPS 186-4 A.1.1.2 に基づく p, q の生成 及び FIPS 186-4 A.2.1 に基づく g の生成	• FIPS 186-4 A.1.1.2 に基づく p, q の生成 及び FIPS 186-4 A.2.1 に基づく g の生成, • FIPS 186-4 A.1.1.2 に基づく p, q の生成 及び FIPS 186-4 A.2.3 に基づく g の生成, • FIPS 186-4 A.1.2.1 に基づく p, q の生成 及び FIPS 186-4 A.2.1 に基づく g の生成, • FIPS 186-4 A.1.2.1 に基づく p, q の生成 及び FIPS 186-4 A.2.3 に基づく g の生成 のいずれか
	FIPS 186-4 A.1.1.2 に基づく p, q の生成	ハッシュ関数識別子	SHA-256	FB の場合, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256 FC の場合, SHA-256, SHA-384, SHA-512, SHA-512/256
		domain_parameter_seed のビット長	256	FB の場合, 224 以上かつ 16000 以下 FC の場合, 256 以上かつ 16000 以下
		生成個数	10	5 以上
	FIPS 186-4 A.1.2.1 に基づく p, q の生成	ハッシュ関数識別子	SHA-256	FB の場合, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256 FC の場合, SHA-256, SHA-384, SHA-512, SHA-512/256
		first_seed のビット長	256	FB の場合, 8 の倍数かつ 224 以上かつ 16000 以下 FC の場合, 8 の倍数かつ 256 以上かつ 16000 以下
		生成する p, q の数	10	5 以上
	FIPS 186-4 A.2.1 に基づく g の生成	生成する g の数	10	5 以上
	FIPS 186-4 A.2.3 に基	*domain_parameter_seed* のビット長	256	• FIPS 186-4 A.1.1.2 に基づく p, q の生成の選択時
次のページに続く	づく g の生成			

前のページからの続き				
試験対象機能	入力欄		既定値	入力条件
				– FB の場合, 8 の倍数かつ 224 以上かつ 16000 以下 – FC の場合, 8 の倍数かつ 256 以上かつ 16000 以下 • FIPS 186-4 A.1.2.1 に基づく p, q の生成の選択時 – FB の場合, 8 の倍数かつ 672 以上かつ 16000 以下 – FC の場合, 8 の倍数かつ 768 以上かつ 16000 以下
		生成する g の数	10	5 以上
ドメインパラメータ検証	FFC パラメータセット		FC	FB, FC のいずれか
	試験の選択		FIPS 186-4 A.1.1.3 に基づく p, q の検証 及び FIPS 186-4 A.2.2 に基づく g の検証	• FIPS 186-4 A.1.1.3 に基づく p, q の検証 及び FIPS 186-4 A.2.2 に基づく g の検証, • FIPS 186-4 A.1.1.3 に基づく p, q の検証 及び FIPS 186-4 A.2.4 に基づく g の検証, • FIPS 186-4 A.1.2.2 に基づく p, q の検証 及び FIPS 186-4 A.2.2 に基づく g の検証, • FIPS 186-4 A.1.2.2 に基づく p, q の検証 及び FIPS 186-4 A.2.4 に基づく g の検証 のいずれか
	FIPS 186-4 A.1.1.3 に基づく p, q の検証	ハッシュ関数識別子	SHA-256	FB の場合, SHA-224,SHA-256,SHA-384,SHA-512 ,SHA-512/224 ,SHA-512/256 FC の場合, SHA-256,SHA-384,SHA-512 ,SHA-512/256
		domain_parameter_seed のビット長	256	FB の場合, 224 以上かつ 16000 以下 FC の場合, 256 以上かつ 16000 以下
		生成個数	10	5 以上
		改ざんするデータの割合 (パーセント)	30	1 以上 99 以下
	FIPS 186-4 A.1.2.2 に基づく p, q の検証	ハッシュ関数識別子	SHA-256	FB の場合, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256 FC の場合, SHA-256, SHA-384, SHA-512, SHA-512/256
		first_seed, *pseed*, *qseed* のビット長	256	FB の場合, 8 の倍数かつ 224 以上かつ 16000 以下 FC の場合, 8 の倍数かつ 256 以上かつ 16000 以下
		生成する p, q の数	10	5 以上
		改ざんするデータの割合 (パーセント)	30	1 以上 99 以下
	FIPS 186-4 A.2.2 に基づく g の検証	生成する g の数	10	5 以上
		改ざんするデータの割合 (パーセント)	30	1 以上 99 以下
次のページに続く				

前のページからの続き				
試験対象機能	入力欄		既定値	入力条件
	FIPS 186-4 A.2.4 に基づく g の検証	*domain_parameter_seed* のビット長	256	• FIPS 186-4 A.1.1.2 に基づく p, q の生成の選択時 – FB の場合, 8 の倍数かつ 224 以上かつ 16000 以下 – FC の場合, 8 の倍数かつ 256 以上かつ 16000 以下 • FIPS 186-4 A.1.2.1 に基づく p, q の生成の選択時 – FB の場合, 8 の倍数かつ 672 以上かつ 16000 以下 – FC の場合, 8 の倍数かつ 768 以上かつ 16000 以下
		生成する g の数	10	5 以上
		改ざんするデータの割合 (パーセント)	30	1 以上 99 以下
鍵ペア生成	FFC パラメータセット		FC	FB, FC のいずれか
	ハッシュ関数識別子		SHA-256	FB の場合, SHA-224,SHA-256,SHA-384,SHA-512 ,SHA-512/224 ,SHA-512/256 FC の場合, SHA-256,SHA-384,SHA-512 ,SHA-512/256
	domain_parameter_seed のビット長		256	FB の場合, 224 以上かつ 16000 以下 FC の場合, 256 以上かつ 16000 以下
	生成個数		10	10 以上
公開鍵検証	FFC パラメータセット		FC	FB, FC のいずれか
	ハッシュ関数識別子		SHA-256	FB の場合, SHA-224,SHA-256,SHA-384,SHA-512 ,SHA-512/224 ,SHA-512/256 FC の場合, SHA-256,SHA-384,SHA-512 ,SHA-512/256
	domain_parameter_seed のビット長		256	FB の場合, 224 以上かつ 16000 以下 FC の場合, 256 以上かつ 16000 以下
	生成個数		10	10 以上
	改ざんするデータの割合 (パーセント)		30	1 以上 99 以下
次のページに続く				

前のページからの続き				
試験対象機能	入力欄		既定値	入力条件
鍵共有	FFC パラメータセット		FC	FB, FC のいずれか
	鍵導出関数		Concatenation KDF with SHA-512	表 4.2 参照
	HMAC を用いる 鍵導出関数の 選択時	*salt* のビット長	256	112 以上 16000 以下の 8 の倍数
	鍵導出関数用 *OtherInfo* データ長		240	16000 以下の 8 の倍数. ただし (IUTid のビット長+JCATT が担う party の *identifier* のビット長) 以上.
	IUTid		a1b2c3d4e5	16 進表記の文字列
	Nonce のビット長		128	FB の場合, 112 以上 16000 以下の 8 の倍数 FC の場合, 128 以上 16000 以下の 8 の倍数
	試験 1	DKM のビット長	鍵導出関数で使用されるハッシュ関数の出力長の 2 倍	8 の倍数かつ 16000 以下
		生成個数	2048	10 以上 10000 以下
		(共有鍵) 改竄するデータの割合 (パーセント)	30	1 以上 99 以下
		検証方法	DKM による検証	DKM による検証 , *Z* による検証 又は MacTag による検証
	試験 2, 試験 3	DKM のビット長	鍵導出関数で使用されるハッシュ関数の出力長の 2 倍	8 の倍数かつ MAC 鍵のビット長以上 16000 以下
		MAC アルゴリズム	HMAC	表 4.3, 表 4.4 を参照.
		MAC アルゴリズムで使用するアルゴリズム	SHA-512	表 4.3, 表 4.4 を参照.
		鍵長 (MAC 鍵のビット長)	256	表 4.3, 表 4.4 を参照.
		MacTag のビット長	256	表 4.3, 表 4.4 を参照.
		Key Confirmation の種類	右に示す選択できる種類で一番上の種類	・dhHybrid1 Unilateral U to V Unilateral V to U Bilateral ・dhHybridOneFlow Unilateral U to V Unilateral V to U Bilateral ・dhStatic Unilateral U to V Unilateral V to U Bilateral ・dhEphem N/A ・dhOneFlow Unilateral V to U
	試験 2	生成個数	2048	10 以上 10000 以下
		改竄するデータの割合 (パーセント)	30	1 以上 99 以下
	試験 3	生成個数	2048	10 以上 10000 以下
		改竄するデータの割合 (パーセント)	30	1 以上 99 以下

表 4.2: Key Establishment Schemes in NIST SP 800-56A で選択可能な鍵導出関数 (FFC)

鍵導出関数の仕様書	鍵導出関数
NIST SP 800-56A	• (Concatenation 又は ASN.1) KDF with SHA-1 • (Concatenation 又は ASN.1) KDF with SHA-224 • (Concatenation 又は ASN.1) KDF with SHA-256 • (Concatenation 又は ASN.1) KDF with SHA-384 • (Concatenation 又は ASN.1) KDF with SHA-512 • (Concatenation 又は ASN.1) KDF with SHA-512/224 • (Concatenation 又は ASN.1) KDF with SHA-512/256 • (Concatenation 又は ASN.1) KDF with HMAC-SHA-1 • (Concatenation 又は ASN.1) KDF with HMAC-SHA-224 • (Concatenation 又は ASN.1) KDF with HMAC-SHA-256 • (Concatenation 又は ASN.1) KDF with HMAC-SHA-384 • (Concatenation 又は ASN.1) KDF with HMAC-SHA-512 • (Concatenation 又は ASN.1) KDF with HMAC-SHA-512/224 • (Concatenation 又は ASN.1) KDF with HMAC-SHA-512/256 • (Concatenation 又は ASN.1) KDF with HMAC-SHA3-256 • (Concatenation 又は ASN.1) KDF with HMAC-SHA3-384 • (Concatenation 又は ASN.1) KDF with HMAC-SHA3-512
ANS X9.42-2001 (ANS X9.63-2001)	• (Concatenation 又は ASN.1) KDF with SHA-1 • (Concatenation 又は ASN.1) KDF with SHA-224 • (Concatenation 又は ASN.1) KDF with SHA-256 • (Concatenation 又は ASN.1) KDF with SHA-384 • (Concatenation 又は ASN.1) KDF with SHA-512
—	無し (shared secret Z による検証の選択時)

表 4.3: “Standard Test Message” を用いた鍵共有の試験及び Key Confirmation で使用する HMAC の条件 (FFC)

FFC パラメータセット	FB	FC
ハッシュ関数	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256 のいずれか	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256 のいずれか
HMAC 鍵のビット長	112 以上 DKM のビット長未満	128 以上 DKM のビット長未満
MacTag のビット長	112 以上 ハッシュ関数の出力長 以下	128 以上 ハッシュ関数の出力長 以下

表 4.4: “Standard Test Message” を用いた鍵共有の試験及び Key Confirmation で使用する CMAC の条件 (FFC)

FFC パラメータセット	FB	FC
ブロック暗号	AES	
MAC 鍵のビット長	128, 192, 256 のいずれかの内, DKM のビット長未満.	
MacTag のビット長	112 以上 128 以下, 8 の倍数	128

4.1.1.2 ECDH in NIST SP800-56A

表 4.5: ECDH in NIST SP800-56A の既定値及び入力条件

試験対象機能	入力欄		既定値	入力条件
ドメインパラメータ生成	p のビット長 (素体上のみ)		256	8 の倍数かつ 16000 以下
	既約多項式の次数 (標数 2 の体上のみ)		283	16000 以下
	生成個数		10	10 以上 10000 以下
ドメインパラメータ検証	なし		–	–
鍵ペア生成	生成個数		10	10 以上
公開鍵検証	生成個数		10	10 以上
	改ざんするデータの割合 (パーセント)		30	1 以上 99 以下
鍵共有	ECC パラメータセット		ドメインパラメータに連動	EB, EC, ED, EE のいずれか
	鍵導出関数		Concatenation KDF with SHA-512	表 4.6 参照
	HMAC を用いる鍵導出関数の選択時	*salt* のビット長	256	112 以上 16000 以下の 8 の倍数
	鍵導出関数用 *OtherInfo* データ長		376	16000 以下の 8 の倍数. ただし (IUTid のビット長+JCATT が担う party の *identifier* のビット長) 以上.
	IUTid		a1b2c3d4e5	16 進表記の文字列
	試験 1	DKM のビット長	鍵導出関数で使用されるハッシュ関数出力の 2 倍のビット長	8 の倍数かつ 16000 以下
		生成個数	2048	10 以上 10000 以下
		改竄するデータの割合 (パーセント)	30	1 以上 99 以下
		検証方法	DKM による検証	DKM による検証 , shared secret Z による検証 又は *MacTag* による検証
	試験 2,	DKM のビット長	鍵導出関数で使用されるハッシュ関数出力の 2 倍のビット長	8 の倍数かつ MAC 鍵のビット長以上 16000 以下
	試験 3,	MAC アルゴリズム	HMAC	表 4.7, 表 4.8 を参照.
		MAC アルゴリズムで使用するアルゴリズム	SHA-512	表 4.7, 表 4.8 を参照.
		鍵長 (MAC 鍵のビット長)	256	表 4.7, 表 4.8 を参照.
		MacTag のビット長	256	表 4.7, 表 4.8 を参照.
		Key Confirmation の種類	右に示す選択できる種類で一番上の種類	・(Cofactor) Full Unified Model Unilateral U to V Unilateral V to U Bilateral ・(Cofactor) One-Pass Unified Model Unilateral U to V Unilateral V to U Bilateral ・(Cofactor) Static Unified Model Unilateral U to V Unilateral V to U Bilateral ・(Cofactor) Ephemeral Unified Model N/A ・(Cofactor) One-Pass Diffie-Hellman Unilateral V to U
	試験 2	生成個数	2048	10 以上 10000 以下

		改竄するデータの割合 (パーセント)	30	1 以上 99 以下
	試験 3	生成個数	2048	10 以上 10000 以下
		改竄するデータの割合 (パーセント)	30	1 以上 99 以下

表 4.6: Key Establishment Schemes in NIST SP 800-56A で選択可能な鍵導出関数 (ECC)

鍵導出関数の仕様書	ドメインパラメータセット		
	EB, EC	ED	EE
	鍵導出関数		
NIST SP 800-56A	• (Concatenation 又は ASN.1) KDF with SHA-1	—	—
	• (Concatenation 又は ASN.1) KDF with HMAC-SHA-1	—	—
	• (Concatenation 又は ASN.1) KDF with SHA-224	同左	—
	• (Concatenation 又は ASN.1) KDF with SHA-512/224		—
	• (Concatenation 又は ASN.1) KDF with HMAC-SHA-224		—
	• (Concatenation 又は ASN.1) KDF with HMAC-SHA-512/224		—
	• (Concatenation 又は ASN.1) KDF with SHA-256		同左
	• (Concatenation 又は ASN.1) KDF with SHA-384		
	• (Concatenation 又は ASN.1) KDF with SHA-512		
	• (Concatenation 又は ASN.1) KDF with SHA-512/256		
	• (Concatenation 又は ASN.1) KDF with HMAC-SHA-256		
	• (Concatenation 又は ASN.1) KDF with HMAC-SHA-384		
	• (Concatenation 又は ASN.1) KDF with HMAC-SHA-512		
	• (Concatenation 又は ASN.1) KDF with HMAC-SHA-512/256		
	• (Concatenation 又は ASN.1) KDF with HMAC-SHA3-256		
	• (Concatenation 又は ASN.1) KDF with HMAC-SHA3-384		
	• (Concatenation 又は ASN.1) KDF with HMAC-SHA3-512		
ANS X9.42-2001 (ANS X9.63-2001)	• (Concatenation 又は ASN.1) KDF with SHA-1	—	—
	• (Concatenation 又は ASN.1) KDF with SHA-224	同左	—
	• (Concatenation 又は ASN.1) KDF with SHA-256		同左
	• (Concatenation 又は ASN.1) KDF with SHA-384		
	• (Concatenation 又は ASN.1) KDF with SHA-512		
—	無し (shared secret Z による検証の選択時)		

表 4.7: “Standard Test Message” を用いた鍵共有の試験及び Key Confirmation で使用する HMAC の条件 (ECC)

ECC パラメータセット	EB	EC	ED	EE
ハッシュ関数	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256 のいずれか	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256 のいずれか	SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256 のいずれか	SHA-256, SHA-384, SHA-512, SHA-512/256 のいずれか
HMAC 鍵のビット長	112 以上 DKM のビット長未満	128 以上 DKM のビット長未満	192 以上 DKM のビット長未満	256 以上 DKM のビット長未満
MacTag のビット長	112 以上 ハッシュ関数の出力長以下	128 以上 ハッシュ関数の出力長以下	192 以上 ハッシュ関数の出力長以下	256 以上 ハッシュ関数の出力長以下

表 4.8: “Standard Test Message” を用いた鍵共有の試験及び Key Confirmation で使用する CMAC の条件 (ECC)

ECC パラメータセット	EB	EC	ED	EE
ブロック暗号	AES		CMAC は使用不可	
MAC 鍵のビット長	128, 192, 256 のいずれかの内, DKM のビット長未満.			
MacTag のビット長	112 以上 128 以下, 8 の倍数	128		

4.1.1.3 素体上 ECDH in SEC1

表 4.9: 素体上 ECDH in SEC1 の既定値及び入力条件

試験対象機能	入力欄	既定値	入力条件
ドメインパラメータ生成	*p* のビット長	256	8 の倍数かつ 16000 以下
	曲線のランダム性検証用 SEED のビット長	256	8 の倍数かつ 16000 以下
	生成個数	10	10 以上
ドメインパラメータ検証	なし	–	–
鍵ペア生成	生成個数	10	10 以上
公開鍵検証	生成個数	10	10 以上
	改ざんするデータの割合 (パーセント)	30	1 以上 99 以下
鍵共有	生成個数	10	10 以上
	鍵導出関数	ANSI X9.63 KDF with SHA-256	ANSI X9.63 KDF with (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 のいずれか)
	OtherInfo のビット長	0	8 の倍数かつ 16000 以下
	共有鍵のビット長	鍵導出関数で使用されるハッシュ関数の出力長	8 の倍数かつ 16000 以下

4.1.1.4 標数 2 の体上 ECDH in SEC1

表 4.10: 標数 2 の体上 ECDH in SEC1 の既定値及び入力条件

試験対象機能	入力欄	既定値	入力条件
ドメインパラメータ生成	既約多項式の次数	283	16000 以下
	曲線のランダム性検証用 SEED のビット長	256	8 の倍数かつ 16000 以下
	生成個数	10	10 以上
ドメインパラメータ検証	なし	–	–
鍵ペア生成	生成個数	10	10 以上
公開鍵検証	生成個数	10	10 以上
	改ざんするデータの割合 (パーセント)	30	1 以上 99 以下
鍵共有	生成個数	10	10 以上
	鍵導出関数	ANSI X9.63 KDF with SHA-256	ANSI X9.63 KDF with (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 のいずれか)
	OtherInfo のビット長	0	8 の倍数かつ 16000 以下
	共有鍵のビット長	鍵導出関数で使用されるハッシュ関数の出力長	8 の倍数かつ 16000 以下

4.1.1.5 Key Establishment Schemes in NIST SP800-56B に共通するパラメータ

表 4.11: Key Establishment Schemes in NIST SP 800-56B で選択可能な鍵導出関数

鍵導出関数の仕様書	鍵導出関数
NIST SP 800-56B	• (Concatenation 又は ASN.1) KDF with SHA-1 • (Concatenation 又は ASN.1) KDF with SHA-224 • (Concatenation 又は ASN.1) KDF with SHA-256 • (Concatenation 又は ASN.1) KDF with SHA-384 • (Concatenation 又は ASN.1) KDF with SHA-512 • (Concatenation 又は ASN.1) KDF with SHA-512/224 • (Concatenation 又は ASN.1) KDF with SHA-512/256 • (Concatenation 又は ASN.1) KDF with SHA3-256 • (Concatenation 又は ASN.1) KDF with SHA3-384 • (Concatenation 又は ASN.1) KDF with SHA3-512 • (Concatenation 又は ASN.1) KDF with HMAC-SHA-1 • (Concatenation 又は ASN.1) KDF with HMAC-SHA-224 • (Concatenation 又は ASN.1) KDF with HMAC-SHA-256 • (Concatenation 又は ASN.1) KDF with HMAC-SHA-384 • (Concatenation 又は ASN.1) KDF with HMAC-SHA-512 • (Concatenation 又は ASN.1) KDF with HMAC-SHA-512/224 • (Concatenation 又は ASN.1) KDF with HMAC-SHA-512/256 • (Concatenation 又は ASN.1) KDF with HMAC-SHA3-256 • (Concatenation 又は ASN.1) KDF with HMAC-SHA3-384 • (Concatenation 又は ASN.1) KDF with HMAC-SHA3-512
ANS X9.42-2001 (ANS X9.63-2001)	• (Concatenation 又は ASN.1) KDF with SHA-1 • (Concatenation 又は ASN.1) KDF with SHA-224 • (Concatenation 又は ASN.1) KDF with SHA-256 • (Concatenation 又は ASN.1) KDF with SHA-384 • (Concatenation 又は ASN.1) KDF with SHA-512

表 4.12: Key Establishment Schemes in NIST SP 800-56B で選択可能な, Key Confirmation に用いる HMAC の条件

鍵長	2048	3072, 4096, 6144, 8192
ハッシュ関数	• SHA-1, • SHA-224, • SHA-256, • SHA-384, • SHA-512, • SHA-512/224, • SHA-512/256, • SHA3-256, • SHA3-384, • SHA3-512	
HMAC 鍵のビット長	112 以上, DKM のビット長未満	128 以上, DKM のビット長未満
MacTag のビット長	64 以上, ハッシュ関数の出力長以下	64 以上, ハッシュ関数の出力長以下

表 4.13: Key Establishment Schemes in NIST SP 800-56B で選択可能な, Key Confirmation に用いる CMAC の条件

鍵長	2048	3072, 4096, 6144, 8192
ブロック暗号	AES	
MAC 鍵のビット長	128, 192, 256 のいずれかの内, DKM のビット長未満	
MacTag のビット長	64 以上 128 以下, 8 の倍数	

4.1.1.6 KAS1 in NIST SP800-56B

表 4.14: KAS1 の既定値及び入力条件

試験対象機能	入力欄			既定値	入力条件
鍵共有 Party U	鍵長			2048	2048, 3072, 4096, 6144, 8192
	公開鍵 e のタイプ選択			65537	65537, ランダム のいずれか
	使用する 鍵導出関数			Concatenation KDF with SHA-256	表 4.11 参照
	HMAC を用いる 鍵導出関数の選択時		*salt* のビット長	256	• 8 の倍数 • 112 以上 • 16000 以下
	鍵導出関数用 *OtherInfo* のビット長			384	• 8 の倍数 • (IUTid のビット長+JCATT が担う party の *identifier* のビット長 +$Nonce_V$ のビット長) 以上 • 16000 以下
	IUTid (or ID_U)			a1b2c3d4e5	16 進表記の文字列
	$Nonce_V$ のビット長			256	鍵長が 2048 の場合, 224 以上 16000 以下の 8 の倍数 鍵長が 3072 の場合, 256 以上 16000 以下の 8 の倍数 鍵長が 4096 の場合, 304 以上 16000 以下の 8 の倍数 鍵長が 6144 の場合, 352 以上 16000 以下の 8 の倍数 鍵長が 8192 の場合, 400 以上 16000 以下の 8 の倍数
	試験 1	DKM のビット長		鍵導出関数で使用される ハッシュ関数の出力長 の 2 倍	• 8 の倍数 • 112 以上 • 16000 以下
		生成個数		2048	10 以上
	試験 2	secret value Z のビット長		鍵長	鍵長に等しい
		DKM のビット長		鍵導出関数で使用される ハッシュ関数の出力長 の 2 倍	• 8 の倍数 • *MacKey* のビット長以上 • 16000 以下
		Key Confirmation に用いる MAC アルゴリズム	MAC アルゴリズム	HMAC	HMAC 又は CMAC
			MAC アルゴリズム で使用する アルゴリズム	SHA-256	表 4.12, 表 4.13 を参照
		MacKey のビット長		128	表 4.12, 表 4.13 を参照
		MacTag のビット長		MAC アルゴリズム の出力長	表 4.12, 表 4.13 を参照
		生成個数		200	10 以上
		改ざんするデータの割合 (パーセント)		50	1 以上 99 以下
鍵共有 Party V	鍵長			2048	2048, 3072, 4096, 6144, 8192
	プライベート鍵のタイプ選択			CRT なし	CRT なし
	使用する 鍵導出関数			Concatenation KDF with SHA-256	表 4.11 参照
	HMAC を用いる 鍵導出関数の選択時		*salt* のビット長	256	• 8 の倍数 • 112 以上 • 16000 以下
	鍵導出関数用 *OtherInfo* のビット長			384	• 8 の倍数 • (IUTid のビット長+JCATT が担う party の *identifier* のビット長 +$Nonce_V$ のビット長) 以上

				• 16000 以下
	IUTid (or ID_V)		a1b2c3d4e5	16 進表記の文字列
	$Nonce_V$ のビット長		256	鍵長が 2048 の場合, 224 以上 16000 以下の 8 の倍数 鍵長が 3072 の場合, 256 以上 16000 以下の 8 の倍数 鍵長が 4096 の場合, 304 以上 16000 以下の 8 の倍数 鍵長が 6144 の場合, 352 以上 16000 以下の 8 の倍数 鍵長が 8192 の場合, 400 以上 16000 以下の 8 の倍数
	試験 1	DKM のビット長	鍵導出関数で使用されるハッシュ関数の出力長の 2 倍	• 8 の倍数 • 112 以上 • 16000 以下
		生成個数	200	10 以上
		改ざんするデータの割合 (パーセント)	50	1 以上 99 以下
	試験 2	暗号文 C のビット長	鍵長	鍵長に等しい
		DKM のビット長	鍵導出関数で使用されるハッシュ関数の出力長の 2 倍	• 8 の倍数 • *MacKey* のビット長以上 • 16000 以下
		Key Confirmation に用いる MAC アルゴリズム — MAC アルゴリズム	HMAC	HMAC 又は CMAC
		Key Confirmation に用いる MAC アルゴリズム — MAC アルゴリズムで使用するアルゴリズム	SHA-256	表 4.12, 表 4.13 を参照
		MacKey のビット長	128	表 4.12, 表 4.13 を参照
		MacTag のビット長	MAC アルゴリズムの出力長	表 4.12, 表 4.13 を参照
		生成個数	200	10 以上

4.1.1.7 KAS2 in NIST SP800-56B

表 4.15: KAS2 の既定値及び入力条件

試験対象機能	入力欄			既定値	入力条件
鍵共有 Party U	Party U	鍵長		2048	2048, 3072, 4096, 6144, 8192
		プライベート鍵のタイプ選択		CRT なし	CRT なし
	Party V	鍵長		2048	2048, 3072, 4096, 6144, 8192
		公開鍵 e のタイプ選択		65537	65537, ランダム のいずれか
	使用する 鍵導出関数			Concatenation KDF with SHA-256	表 4.11 参照
	HMAC を用いる 鍵導出関数の選択時		*salt* のビット長	256	• 8 の倍数 • 112 以上 • 16000 以下
	鍵導出関数用 *OtherInfo* のビット長			384	• 8 の倍数 • (IUTid のビット長+JCATT が担う party の *identifier* のビット長) 以上 • 16000 以下
	IUTid (or ID_U)			a1b2c3d4e5	16 進表記の文字列
	試験 1	DKM のビット長		鍵導出関数で使用される ハッシュ関数の出力長 の 2 倍	• 8 の倍数 • 112 以上 • 16000 以下
		生成個数		2048	10 以上
		暗号文 C_U の数 (random generation test)		200	10 以上
	試験 2	DKM のビット長		鍵導出関数で使用される ハッシュ関数の出力長 の 2 倍	• 8 の倍数 • *MacKey* のビット長以上 • 16000 以下
		Key Confirmation に用いる MAC アルゴリズム	MAC アルゴリズム	HMAC	HMAC 又は CMAC
			MAC アルゴリズム で使用する アルゴリズム	SHA-256	表 4.12, 表 4.13 を参照
		MacKey のビット長		128	表 4.12, 表 4.13 を参照
		MacTag のビット長		MAC アルゴリズム の出力長	表 4.12, 表 4.13 を参照
		Key Confirmation Provider の試験	生成個数	200	10 以上
		Key Confirmation Recipient の試験	生成個数	200	10 以上
			改ざんするデータの割合 (パーセント)	50	1 以上 99 以下
	試験 3	DKM のビット長		鍵導出関数で使用される ハッシュ関数の出力長 の 2 倍	• 8 の倍数 • *MacKey* のビット長以上 • 16000 以下
		Key Confirmation に用いる MAC アルゴリズム	MAC アルゴリズム	HMAC	HMAC 又は CMAC
			MAC アルゴリズム で使用する アルゴリズム	SHA-256	表 4.12, 表 4.13 を参照
		MacKey のビット長		128	表 4.12, 表 4.13 を参照
		MacTag のビット長		MAC アルゴリズム の出力長	表 4.12, 表 4.13 を参照
		生成個数		200	10 以上
		改ざんするデータの割合 (パーセント)		50	1 以上 99 以下
	Party U	鍵長		2048	2048, 3072, 4096, 6144, 8192
		公開鍵 e のタイプ選択		65537	65537, ランダム のいずれか

鍵共有 Party V	Party V	鍵長		2048	2048, 3072, 4096, 6144, 8192
		プライベート鍵のタイプ選択		CRT なし	CRT なし
	使用する 鍵導出関数			Concatenation KDF with SHA-256	表 4.11 参照
	HMAC を用いる 鍵導出関数の選択時		*salt* のビット長	256	• 8 の倍数 • 112 以上 • 16000 以下
	鍵導出関数用 *OtherInfo* のビット長			384	• 8 の倍数 • (IUTid のビット長+JCATT が担う party の *identifier* のビット長) 以上 • 16000 以下
	IUTid (or ID_V)			a1b2c3d4e5	16 進表記の文字列
	試験 1	DKM のビット長		鍵導出関数で使用される ハッシュ関数の出力長 の 2 倍	• 8 の倍数 • 112 以上 • 16000 以下
		生成個数		2048	10 以上
		暗号文 C_V の数 (random generation test)		200	10 以上
	試験 2	DKM のビット長		鍵導出関数で使用される ハッシュ関数の出力長 の 2 倍	• 8 の倍数 • *MacKey* のビット長以上 • 16000 以下
		Key Confirmation に用いる MAC アルゴリズム	MAC アルゴリズム	HMAC	HMAC 又は CMAC
			MAC アルゴリズム で使用する アルゴリズム	SHA-256	表 4.12, 表 4.13 を参照
		MacKey のビット長		128	表 4.12, 表 4.13 を参照
		MacTag のビット長		MAC アルゴリズム の出力長	表 4.12, 表 4.13 を参照
		Key Confirmation Provider の試験	生成個数	200	10 以上
			改ざんするデータの割合 (パーセント)	50	1 以上 99 以下
		Key Confirmation Recipient の試験	生成個数	200	10 以上
	試験 3	DKM のビット長		鍵導出関数で使用される ハッシュ関数の出力長 の 2 倍	• 8 の倍数 • *MacKey* のビット長以上 • 16000 以下
		Key Confirmation に用いる MAC アルゴリズム	MAC アルゴリズム	HMAC	HMAC 又は CMAC
			MAC アルゴリズム で使用する アルゴリズム	SHA-256	表 4.12, 表 4.13 を参照
		MacKey のビット長		128	表 4.12, 表 4.13 を参照
		MacTag のビット長		MAC アルゴリズム の出力長	表 4.12, 表 4.13 を参照
		生成個数		200	10 以上
		改ざんするデータの割合 (パーセント)		50	1 以上 99 以下

4.1.1.8 KTS-OAEP in NIST SP800-56B

表 4.16: KTS-OAEP の既定値及び入力条件

試験対象機能	入力欄			既定値	入力条件
暗号化	鍵長			2048	2048, 3072, 4096, 6144, 8192
	公開鍵 *e* のタイプ選択			65537	65537, ランダム のいずれか
	ハッシュ関数			SHA-256	表 4.17 参照
	マスク生成関数 *MGF*			MGF with SHA-256	表 4.18 参照
	additional input *A* のビット長			64	8 の倍数かつ 16000 以下
	試験 1	keying material *K* のビット長		256	8 の倍数かつ (鍵長 - ハッシュ関数の出力長の 2 倍 - 16) 以下
		keying material *K* の数		2048	10 以上
		暗号文の数 (random generation test)		10	10 以上
	試験 2	*mgfSeed* のビット長		ハッシュ関数の出力長	ハッシュ関数の出力長に等しい
		keying material *K* のビット長		256	8 の倍数かつ (鍵長 - ハッシュ関数の出力長の 2 倍 - 16) 以下
		生成個数		200	10 以上
	試験 3	keying material *K* のビット長		256	• 8 の倍数 • *MacKey* のビット長以上 • (鍵長 - ハッシュ関数の出力長の 2 倍 - 16) 以下
		Key Confirmation に用いる MAC アルゴリズム	MAC アルゴリズム	HMAC	HMAC 又は CMAC
			MAC アルゴリズムで使用するアルゴリズム	SHA-256	表 4.12, 表 4.13 を参照
		MacKey のビット長		128	表 4.12, 表 4.13 を参照
		MacTag のビット長		MAC アルゴリズムの出力長	表 4.12, 表 4.13 を参照
		生成個数		200	10 以上
		改ざんするデータの割合 (パーセント)		50	1 以上 99 以下
復号	鍵長			2048	2048, 3072, 4096, 6144, 8192
	プライベート鍵のタイプ選択			CRT なし	CRT なし
	ハッシュ関数			SHA-256	表 4.17 参照
	マスク生成関数 *MGF*			MGF with SHA-256	表 4.18 参照
	additional input *A* のビット長			64	8 の倍数かつ 16000 以下
	試験 1	keying material *K* のビット長		256	8 の倍数かつ (鍵長 - ハッシュ関数の出力長の 2 倍 - 16) 以下
		暗号文の数		200	10 以上
		改ざんするデータの割合 (パーセント)		50	1 以上 99 以下
	試験 2	keying material *K* のビット長		256	• 8 の倍数 • *MacKey* のビット長以上 • (鍵長 - ハッシュ関数の出力長の 2 倍 - 16) 以下
		Key Confirmation に用いる MAC アルゴリズム	MAC アルゴリズム	HMAC	HMAC 又は CMAC
			MAC アルゴリズムで使用するアルゴリズム	SHA-256	表 4.12, 表 4.13 を参照
		MacKey のビット長		128	表 4.12, 表 4.13 を参照
		MacTag のビット長		MAC アルゴリズムの出力長	表 4.12, 表 4.13 を参照
		暗号文の数		200	10 以上
		改ざんするデータの割合 (パーセント)		50	1 以上 99 以下

表 4.17: KTS-OAEP in NIST SP 800-56B で選択可能なハッシュ関数

ハッシュ関数の仕様書	選択可能なハッシュ関数
FIPS 180-4	• SHA-1 • SHA-224 • SHA-256 • SHA-384 • SHA-512 • SHA-512/224 • SHA-512/256
FIPS 202	• SHA3-256 • SHA3-384 • SHA3-512

表 4.18: KTS-OAEP in NIST SP 800-56B で選択可能なマスク生成関数 *MGF*

マスク生成関数の仕様書	マスク生成関数 *MGF* の中で使用可能なハッシュ関数
NIST SP800-56B	• SHA-1 • SHA-224 • SHA-256 • SHA-384 • SHA-512 • SHA-512/224 • SHA-512/256 • SHA3-256 • SHA3-384 • SHA3-512

4.1.1.9 NIST SP800-56B の要素機能

表 4.19: NIST SP800-56B で規定された要素機能の既定値及び入力条件

試験対象要素機能	入力欄	既定値	入力条件
鍵ペア生成	鍵長	2048	2048, 3072, 4096, 6144, 8192
	プライベート鍵のタイプ選択	CRT あり	CRT あり, CRT なし のいずれか
	公開鍵 *e* のタイプ選択	65537	65537, ランダム のいずれか
	鍵の個数	10	10 以上
公開鍵部分検証	鍵長	2048	2048, 3072, 4096, 6144, 8192
	公開鍵 *e* のタイプ選択	65537	65537, ランダム のいずれか
	鍵の個数	20	20 以上
	改ざんするデータの割合 (パーセント)	50	1 以上 99 以下
RSAEP	鍵長	2048	2048, 3072, 4096, 6144, 8192
	公開鍵 *e* のタイプ選択	65537	65537, ランダム のいずれか
	integer *k* のビット長	鍵長	鍵長に等しい
	暗号文の数	2048	10 以上
RSADP	鍵長	2048	2048, 3072, 4096, 6144, 8192
	プライベート鍵のタイプ選択	CRT なし	CRT なし
	暗号文 *c* のビット長	鍵長	鍵長に等しい
	生成個数	200	10 以上
	改ざんするデータの割合 (パーセント)	50	1 以上 99 以下
RSASVE Generate	鍵長	2048	2048, 3072, 4096, 6144, 8192
	公開鍵 *e* のタイプ選択	65537	65537, ランダム のいずれか
	secret value *Z* のビット長	鍵長	鍵長に等しい
	暗号文の数	2048	10 以上
RSASVE Recover	鍵長	2048	2048, 3072, 4096, 6144, 8192
	プライベート鍵のタイプ選択	CRT なし	CRT なし
	暗号文 *C* のビット長	鍵長	鍵長に等しい
	生成個数	200	10 以上
	改ざんするデータの割合 (パーセント)	50	1 以上 99 以下

4.1.1.10 PSEC-KEM

表 4.20: PSEC-KEM の既定値及び入力条件 (素体上と標数 2 の体上とで共通)

試験対象機能	入力欄	既定値	入力条件
鍵ペア生成	生成個数	10	10 以上
セッション鍵暗号化	暗号文の数	10	10 以上
	鍵導出関数	ISO/IEC 18033-2 KDF1 with SHA-256	ISO/IEC 18033-2 KDF1 with (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 のいずれか)
	セッション鍵の長さ	128	8 の倍数かつ 16000 以下
	乱数の長さ	512	8 の倍数かつ 16000 以下
	マスク生成時の点形式	uncompressed	uncompressed, compressed, hybrid のいずれか
セッション鍵復号	暗号文の数	10	10 以上
	鍵導出関数	ISO/IEC 18033-2 KDF1 with SHA-256	ISO/IEC 18033-2 KDF1 with (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 のいずれか)
	セッション鍵の長さ	128	8 の倍数かつ 16000 以下
	乱数の長さ	512	8 の倍数かつ 16000 以下
	マスク生成時の点形式	uncompressed	uncompressed, compressed, hybrid のいずれか
	改ざんするデータの割合 (パーセント)	30	1 以上 99 以下

4.1.2 鍵導出関数

4.1.2.1 KDF in NIST SP800-108

4.1.2.1.1 KDF in Counter Mode, KDF in Feedback Mode, KDF in Double-Pipeline Iteration Mode 共通

表 4.21: 擬似ランダム関数 (PRF) と鍵導出鍵の長さ
(KDF in Counter Mode, KDF in Feedback Mode, KDF in Double-Pipeline Iterantion Mode 共通)

試験対象機能	入力欄		既定値	入力条件
鍵導出関数	使用する PRF	MAC アルゴリズム	HMAC	HMAC 又は CMAC
		MAC アルゴリズムの中で使用するアルゴリズム	SHA-256	HMAC を選択した場合, 次の暗号アルゴリズムから選択可能 ● SHA-1 ● SHA-224 ● SHA-256 ● SHA-384 ● SHA-512 ● SHA-512/224 ● SHA-512/256 ● SHA3-256 ● SHA3-384 ● SHA3-512 CMAC を選択した場合, ● AES ● 3-key Triple DES
	鍵導出鍵 K_I のビット長		256	HMAC を選択時, ● 8 の倍数 ● 112 以上 ● 16000 以下 AES を使用した CMAC を選択時, 128, 192, 256 のいずれか 3-key Triple DES を使用した CMAC を選択時, 168

4.1.2.1.2 KDF in Counter Mode

表 4.22: 鍵導出関数の既定値及び入力条件 (KDF in Counter Mode)

試験対象機能	入力欄		既定値	入力条件
鍵導出関数	counter i の長さ (ビット数) r		32	8, 16, 24 又は 32
	使用する PRF		HMAC-SHA-256	表 4.21 参照
	鍵導出鍵 K_I のビット長		256	表 4.21 参照
	IUT がサポートする L の内 h で割り切れる最小値		256	• h の倍数 • h 以上 • 最大値以下
	IUT がサポートする L の内 h で割り切れる最大値		1024	• h の倍数 • 最小値以上 • 4096 以下
	h で割り切れない L を IUT がサポートする場合	IUT がサポートする L の内 h で割り切れない最小値	112	• 8 の倍数 • 112 以上 • 最大値以下
		IUT がサポートする L の内 h で割り切れない最大値	640	• 8 の倍数 • 最小値以上 • 4096 以下
	生成個数		10	10 以上

4.1.2.1.3 KDF in Feedback Mode

表 4.23: 鍵導出関数の既定値及び入力条件 (KDF in Feedback Mode)

試験対象機能	入力欄		既定値	入力条件
鍵導出関数	counter i の長さ (ビット数) r		32	8, 16, 24 又は 32
	counter を入力に含めるか否かの選択		含める	含める, 含めないのいずれか
	使用する PRF		HMAC-SHA-256	表 4.21 参照
	鍵導出鍵 K_I のビット長		256	表 4.21 参照
	IUT がサポートする L の内 h で割り切れる最小値		256	• h の倍数 • h 以上 • 最大値以下
	IUT がサポートする L の内 h で割り切れる最大値		1024	• h の倍数 • 最小値以上 • 4096 以下
	h で割り切れない L を IUT がサポートする場合	IUT がサポートする L の内 h で割り切れない最小値	112	• 8 の倍数 • 112 以上 • 最大値以下
		IUT がサポートする L の内 h で割り切れない最大値	640	• 8 の倍数 • 最小値以上 • 4096 以下
	IUT がサポートする $\mathbf{len}(IV)$	0	0	0
		$0 < \mathbf{len}(IV) < h$	$h/2$	• 8 の倍数 • 8 以上 • h に満たない
		$\mathbf{len}(IV) = h$	h	• h に等しい
		$\mathbf{len}(IV) > h$	$\frac{3}{2}h$	• 8 の倍数 • h を上回る
	生成個数		10	10 以上

4.1.2.1.4 KDF in Double-Pipeline Iteration Mode

表 4.24: 鍵導出関数の既定値及び入力条件 (KDF in Double-Pipeline Iterantion Mode)

試験対象機能	入力欄		既定値	入力条件
鍵導出関数	counter i の長さ (ビット数) r		32	8, 16, 24 又は 32
	counter を入力に含めるか否かの選択		含める	含める, 含めないのいずれか
	使用する PRF		HMAC-SHA-256	表 4.21 参照
	鍵導出鍵 K_I のビット長		256	表 4.21 参照
	IUT がサポートする L の内 h で割り切れる最小値		256	• h の倍数 • h 以上 • 最大値以下
	IUT がサポートする L の内 h で割り切れる最大値		1024	• h の倍数 • 最小値以上 • 4096 以下
	h で割り切れない L を IUT がサポートする場合	IUT がサポートする L の内 h で割り切れない最小値	112	• 8 の倍数 • 112 以上 • 最大値以下
		IUT がサポートする L の内 h で割り切れない最大値	640	• 8 の倍数 • 最小値以上 • 4096 以下
	生成個数		10	10 以上

4.1.2.2 PBKDF in NIST SP800-132

表 4.25: PBKDF の既定値及び入力条件

試験対象機能	入力欄		既定値	入力条件
鍵導出関数	使用する PRF		HMAC-SHA-256	次の暗号アルゴリズムから選択可能 • HMAC-SHA-1 • HMAC-SHA-224 • HMAC-SHA-256 • HMAC-SHA-384 • HMAC-SHA-512 • HMAC-SHA-512/224 • HMAC-SHA-512/256 • HMAC-SHA3-256 • HMAC-SHA3-384 • HMAC-SHA3-512
	Iteration count C		1000	• 1,000 以上 • 100,000 以下
	Password P のビット長	$112 \leq \mathbf{len}(P) < B$	112	• 8 の倍数 • 112 以上 • B 未満
		$\mathbf{len}(P) = B$	B	
		$\mathbf{len}(P) > B$	$2B$	• 8 の倍数 • B より大きく • 16000 以下
	Salt S のビット長	$128 \leq \mathbf{len}(S) < (hLen - 32)$	128	• 8 の倍数 • 128 以上 • $(hLen - 32)$ 以下
		$\mathbf{len}(S) = (hLen - 32)$	$hLen - 32$	$(hLen - 32)$
		$\mathbf{len}(S) > (hLen - 32)$	$hLen$	• 8 の倍数 • $(hLen - 32)$ より大きく • 16000 以下
	IUT がサポートする $kLen$ の内 $hLen$ で割り切れる最小値		256	• $hLen$ の倍数 • $hLen$ 以上 • 最大値以下
	IUT がサポートする $kLen$ の内 $hLen$ で割り切れる最大値		4096	• $hLen$ の倍数 • 最小値以上 • 8192 以下
	$hLen$ で割り切れない $kLen$ を IUT がサポートする場合	IUT がサポートする $kLen$ の内, $hLen$ で割り切れない最小値	112	• 8 の倍数 • 112 以上 • 最大値以下
		IUT がサポートする $kLen$ の内, $hLen$ で割り切れない最大値	640	• 8 の倍数 • 最小値以上 • 4096 以下
	生成個数		10	10 以上

4.1.2.3 KDF in NIST SP800-135

4.1.2.3.1 IKE version 1

表 4.26: 鍵導出関数の既定値及び入力条件

試験対象機能	入力欄		既定値	入力条件
鍵導出関数	使用する PRF	MAC アルゴリズム	HMAC	HMAC
		MAC アルゴリズムの中で使用するアルゴリズム	SHA-256	次の暗号アルゴリズムから選択可能 • SHA-1 • SHA-256 • SHA-384 • SHA-512
	nonce (*Ni_b*, *Nr_b*) のビット長の最小値		256	• 8 の倍数 • 256 以上 • 最大値以下
	nonce (*Ni_b*, *Nr_b*) のビット長の最大値		2048	• 8 の倍数 • 最小値以上 • 2048 以下
	cookie (*CKY-I*, *CKY-R*) のビット長		64	64 固定
	shared secret (g^{xy}) のビット長		1024	次から選択可能 • 224 • 256 • 384 • 512 • 1024 • 1536 • 2048 • 3072 • 4096 • 6144 • 8192
	IUT が事前共有鍵をサポートする場合	事前共有鍵のビット長の最小値	256	• 8 の倍数 • 112 以上 • 最大値以下
		事前共有鍵のビット長の最大値	512	• 8 の倍数 • 最小値以上 • 4096 以下
	生成個数		10	10 以上

4.1.2.3.2 IKE version 2

表 4.27: 鍵導出関数の既定値及び入力条件

試験対象機能	入力欄		既定値	入力条件
鍵導出関数	使用する PRF	MAC アルゴリズム	HMAC	HMAC
		MAC アルゴリズムの中で使用するアルゴリズム	SHA-256	次の暗号アルゴリズムから選択可能 • SHA-1 • SHA-256 • SHA-384 • SHA-512
	nonce (*Ni*, *Nr*) のビット長の最小値		256	• 8 の倍数 • 128 以上 • 最大値以下
	nonce (*Ni*, *Nr*) のビット長の最大値		2048	• 8 の倍数 • 最小値以上 • 2048 以下
	Security Parameter Index (*SPIi*, *SPIr*) のビット長		64	64 固定
	shared secret (g^{ir}, g^{ir}(new)) のビット長		2048	次から選択可能 • 224 • 256

			• 384 • 512 • 1024 • 1536 • 2048 • 3072 • 4096 • 6144 • 8192
	keying material (*DKM*) のビット長	1280	• 8 の倍数 • *h* 以上 • *h* の 256 倍以下
	Child SA 用の keying material (*DKM*) のビット長	1280	• 8 の倍数 • *h* 以上 • *h* の 256 倍以下
	生成個数	10	10 以上

4.1.2.3.3 Key Derivation in TLS versions 1.0 and 1.1

表 4.28: 鍵導出関数の既定値及び入力条件

試験対象機能	入力欄		既定値	入力条件
鍵導出関数	*ServerHello.Random* のビット長		256	256
	ClientHello.Random のビット長		256	256
	server_random のビット長		256	256
	client_random のビット長		256	256
	試験 1	*pre_master_secret* のビット長	384	次から選択可能 • 224 • 256 • 384 • 512 • 1024 • 1536 • 2048 • 3072 • 4096 • 6144 • 8192
	試験 2	*Z* のビット長	384	次から選択可能 • 224 • 256 • 384 • 512 • 1024 • 1536 • 2048 • 3072 • 4096 • 6144 • 8192
	master_secret のビット長		384	384
	key_block のビット長		1024	384
	試験 1	生成個数	10	10 以上
	試験 2	生成個数	2048	10 以上

4.1.2.3.4 Key Derivation in TLS version 1.2

表 4.29: 鍵導出関数の既定値及び入力条件

試験対象機能	入力欄		既定値	入力条件
鍵導出関数	使用する PRF	MAC アルゴリズム	HMAC	HMAC
		MAC アルゴリズムの中で使用するアルゴリズム	SHA-256	次の暗号アルゴリズムから選択可能 • SHA-256 • SHA-384 • SHA-512
	ServerHello.Random のビット長		256	256
	ClientHello.Random のビット長		256	256
	server_random のビット長		256	256
	client_random のビット長		256	256
	試験 1	*pre_master_secret* のビット長	384	次から選択可能 • 224 • 256 • 384 • 512 • 1024 • 1536 • 2048 • 3072 • 4096 • 6144 • 8192
	試験 2	*Z* のビット長	384	次から選択可能 • 224 • 256 • 384 • 512 • 1024 • 1536 • 2048 • 3072 • 4096 • 6144 • 8192
	master_secret のビット長		384	384
	key_block のビット長		1024	384
	試験 1	生成個数	10	10 以上
	試験 2	生成個数	2048	10 以上

4.1.2.3.5 Key Derivation Functions in ANS X9.42-2001 and ANS X9.63-2001

表 4.30: 鍵導出関数の既定値及び入力条件

試験対象機能	入力欄	既定値	入力条件
鍵導出関数	使用するハッシュ関数	SHA-256	次の暗号アルゴリズムから選択可能 • SHA-1 • SHA-224 • SHA-256 • SHA-384 • SHA-512
	OtherInfo (ANS X9.63 の場合は *SharedInfo*) のビット長の最小値	0	• 8 の倍数 • 0 以上 • 7776 以下 • *OtherInfo* のビット長の最大値以下

	OtherInfo (ANS X9.63 の場合は *SharedInfo*) のビット長の最大値	192	• 8 の倍数 • 0 以上 • 7776 以下 • *OtherInfo* のビット長の最小値以上
	shared secret (*Z*) のビット長	2048	次から選択可能 • 224 • 256 • 384 • 512 • 1024 • 1536 • 2048 • 3072 • 4096 • 6144 • 8192
	keying material (*KeyMat*) のビット長	鍵導出関数で使用されるハッシュ関数の出力長の 2 倍	• 8 の倍数 • 112 以上 • 16000 以下
	生成個数	10	10 以上

4.1.2.3.6 SSH Key Derivation Function

表 4.31: 鍵導出関数の既定値及び入力条件

試験対象機能	入力欄	既定値	入力条件
鍵導出関数	使用するハッシュ関数	SHA-256	次の暗号アルゴリズムから選択可能 • SHA-1 • SHA-224 • SHA-256 • SHA-384 • SHA-512
	shared secret (*K*) のビット長	2048	次から選択可能 • 1024 • 2048
	H のビット長	256	鍵導出関数で使用されるハッシュ関数の出力長
	session_id のビット長	256	鍵導出関数で使用されるハッシュ関数の出力長
	InitialIV のビット長	128	次から選択可能 • 64 • 128
	Encryptionkey のビット長	128	• 8 の倍数 • 112 以上 • 256 以下
	Integritykey のビット長	256	鍵導出関数で使用されるハッシュ関数の出力長
	生成個数	10	10 以上

4.1.2.3.7 SRTP Key Derivation Function

表 4.32: 鍵導出関数の既定値及び入力条件

試験対象機能	入力欄	既定値	入力条件
鍵導出関数	使用するブロック暗号	AES-128	次の暗号アルゴリズムから選択可能

			• AES-128 • AES-192 • AES-256
	master key (*k_master*) のビット長	128	次から選択可能 • 128 • 192 • 256
	master salt (*master_salt*) のビット長	112	112 固定
	key-derivation rate (*kdr*) のビット長	48	48 固定
	SRTP 用 *index* のビット長	48	48 固定
	SRTCP 用 *index* のビット長	48	48 固定
	セッション暗号化鍵 (*k_e*) のビット長	128	次から選択可能 • 128 • 192 • 256
	セッションメッセージ認証鍵 (*k_a*) のビット長	160	160 固定
	セッションソルト鍵 (*k_s*) のビット長	112	112 固定
	生成個数	10	10 以上

4.1.2.3.8 SNMP Key Derivation Function

表 4.33: 鍵導出関数の既定値及び入力条件

試験対象機能	入力欄	既定値	入力条件
鍵導出関数	使用するハッシュ関数	SHA-1	次の暗号アルゴリズムから選択可能 • SHA-1
	password のバイト数	8	8 以上
	snmpEngineID のビット長	96	96 以上
	key のビット長	160	鍵導出関数で使用される ハッシュ関数の出力長
	生成個数	10	10 以上

附 則
この手順は, 平成 21 年 1 月 23 日から施行し, 平成 21 年 1 月 8 日から適用する.

附 則
この手順は, 平成 21 年 7 月 1 日から施行し, 平成 21 年 7 月 10 日から適用する.

附 則
この手順は, 平成 24 年 2 月 29 日から施行し, 平成 24 年 6 月 1 日から適用する.

附 則
この手順は, 平成 30 年 6 月 22 日から施行し, 平成 30 年 6 月 22 日から適用する.

附 則
この手順は, 令和元年 7 月 11 日から施行し, 令和元年 7 月 11 日から適用する.

参考文献

[1] American Bankers Association, *Public Key Cryptography for the Financial Services Industry : Agreement of Symmetric Keys Using Discrete Logarithm Cryptography*, ANS X9.42-2003, November 19, 2003.

[2] American Bankers Association, *Public Key Cryptography For The Financial Services Industry : Key Agreement and Key Transport Using Elliptic Curve Cryptography*, ANSI X9.63-2001, November 20, 2001.

[3] ISO/IEC 18033-2:2006, *Information technology – Security techniques – Encryption algorithms – Part 2: Asymmetric ciphers*.

[4] National Institute of Standards and Technology, *Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography*, NIST SP 800-56A Revision 3, April 2018.

[5] National Institute of Standards and Technology, *Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography*, NIST SP 800-56B Revision 2, March 2019.

[6] National Institute of Standards and Technology, *Recommendation for ObtainingAssurances for Digital SignatureApplications*, NIST SP 800-89, November 2006.

[7] FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION, *Digital Signature Standard (DSS)*, FIPS PUB 186-4, July 2013.

[8] National Institute of Standards and Technology, *Recommendation for Key Derivation Using Pseudorandom Functions (Revised),* NIST SP 800-108, October 2009.

[9] National Institute of Standards and Technology, *Recommendation for Password-Based Key Derivation*, NIST SP 800-132, December 2010.

[10] National Institute of Standards and Technology, *Recommendation for Existing Application-Specific Key Derivation Functions,* NIST SP 800-135 Revision 1, December 2011.

[11] Timothy A. Hall, *The NIST SP 800-135 Existing Application-Specific Key Derivation Function Validation System (ASKDFVS),* National Institute of Standards and Technology, September 5, 2013.

[12] 128 ビットブロック暗号 Camellia アルゴリズム仕様書 (第 2 版: 2001 年 9 月 26 日) https://www.cryptrec.go.jp/cryptrec_03_spec_cypherlist_files/PDF/06_01jspec.pdf

[13] IPA, 暗号アルゴリズム実装試験仕様書 — 公開鍵 — (ATR-01-A), https://www.ipa.go.jp/security/jcmvp/documents/atr/atr01a.pdf

[14] IPA, JCATT ファイルフォーマット仕様書 — 鍵確立手法 —, https://www.ipa.go.jp/security/jcmvp/documents/open/jcatt/format/jcatt_fileformat_f.zip

[15] IPA, JCATT サンプルファイル — 鍵確立手法 —, https://www.ipa.go.jp/security/jcmvp/documents/open/jcatt/sample/jcatt_sample_f.zip

改版履歴

識別番号	ATR-01-F

改訂年月日	作成者・承認者	改訂内容
平成 21 年 1 月 23 日	橋本・仲田	新規制定
平成 21 年 7 月 1 日	櫻井・仲田	一部改正 (DH 及び ECDH の共有鍵のビット長を 鍵導出関数で使用される ハッシュ関数の出力長に修正)
平成 24 年 2 月 29 日	橋本・仲田	一部改正 (NIST SP800-56A に記載の DH 及び ECDH の試験仕様を追加)
平成 30 年 6 月 22 日	櫻井・江口	一部改正 (NIST SP800-56A に記載の DH 及び ECDH の試験仕様を NIST SP800-56A Rev.2 対応に更新. NIST SP800-56B に記載の鍵確立手法, NIST SP800-108 に記載の KDF, NIST SP800-132 に記載の KDF 及び NIST SP800-135 に記載の KDF を追加.)
令和元年 7 月 11 日	櫻井・江口	一部改正 (依存関係のある暗号アルゴリズムを記載及び誤植を訂正. NIST SP800-56B Rev.2 対応に更新.)