## 企業・個人の情報セキュリティ対策事業

# 暗号アルゴリズム実装試験ツールの機能追加

**開発成果概要資料**

**三菱電機株式会社**

**概要**

独立行政法人 情報処理推進機構(IPA)では，暗号モジュール試験及び認証制度(JCMVP: Japan Cryptographic Module Validation Program)の試行運用を 2006 年度から開始し，2007 年度からは本運用を開始した．JCMVP で実施する暗号モジュール試験は, 暗号アルゴリズム実装の試験とその他の試験に大別される. 我々は, JCMVP の暗号アルゴリズム実装試験において使用するための暗号アルゴリズム実装試験ツール(JCATT: Japan Cryptographic Testing Tool)を 2006 年度までに開発した. 今年度は, JCATT がサポートする暗号アルゴリズムの追加等, JCATT の機能追加の開発を行った．本報告書では，開発した JCATT の概要を述べる．

## 1. はじめに

近年，CRYPTREC や NESSIE など様々な機関において暗号アルゴリズムの評価が実施され, 推奨暗号のリストが作成された. 各機関では，各国の第一線の暗号研究者が評価作業に携わってきた．専門家による暗号の安全性と実装性に関する厳しい評価による“お墨付き”を与えられた暗号は，電子政府を含めた各種情報セキュリティシステムにおいて使用が進んでいる.

各暗号評価プロジェクトは，暗号の“アルゴリズム”に関する評価を行うものであった．アルゴリズムが優れていると判定された暗号を使用することは，安全で信頼できる情報セキュリティシステムを構築する上での第一の要件である．しかし，暗号アルゴリズムが実際に使用される時には，ソフトウエアやハードウエアで実装された“暗号モジュール”という形態になることに注意しなければならない．暗号モジュールは，暗号アルゴリズムの仕様書を元に実際に運用するシステムに合わせて実装されるものである．暗号アルゴリズムと暗号モジュールとの間には，暗号モジュール開発者の手作業が入るものであり，暗号モジュール開発者には相応のスキルを有していることが要求される．暗号アルゴリズムの実装には，ソフトウエアやハードウエアの実装スキルだけでなく，数学的スキルも要求されるため，実装は易しい課題ではない.

正しく実装されていない暗号モジュールを使用すれば，情報セキュリティシステムの安全性と信頼性が破綻することになる. すなわち，優れた暗号を使用することと，優れた情報セキュリティシステムを構築す

ることとは等価ではなく，暗号モジュールが暗号アルゴリズムを正しく実装したものであるかどうかの検証が欠かせない．暗号モジュール評価法を規定した FIPS 140-2 の中でも暗号アルゴリズム実装の試験の必要性が述べられている．

最近数年間，特に 2004 年度中に，NIST から FIPS に記述されたいくつかの暗号アルゴリズムに対して暗号アルゴリズム試験方法が公開された[2〜12]．しかし，日本国内メーカによる提案の多い電子政府推奨暗号は，多くの暗号について試験方法が定められていなかった．また，NIST から公開されている試験方法は，ごく単純な試験を行うだけという問題があった．

我々は，NIST の暗号アルゴリズム試験方法も調査した上で，暗号アルゴリズム試験方法を考察し，JCATT の開発を行った．本開発により，電子政府推奨暗号をはじめ，多くの暗号アルゴリズムに対して実装試験が可能となった．

## 2. 暗号アルゴリズム試験方法

### 2.1. NIST の動向

NIST から次の暗号アルゴリズムの試験方法が公開された．

- ブロック暗号：AES[2], Triple DES[11], DES[10]，Skipjack[10]
- ブロック暗号の利用モード：CCM mode of operation[3]
- ハッシュ関数[8]：SHA-1，SHA-224，SHA-256，SHA-384，SHA-512
- メッセージ認証(鍵付きハッシュ関数)：HMAC[6]
- ディジタル署名：RSASSA-PKCS1-v1_5[9], RSA-PSS[9]，DSA[7]，ECDSA[5]
- 擬似乱数生成：ANSI 9.31 の方法[7]

本開発で検討した暗号アルゴリズム試験方法と，開発した JCATT は，NIST の試験方法をカバーし，さらにより詳細な試験を行っている．

### 2.2. 追加した暗号アルゴリズム

JCATT に今年度追加した試験対象暗号アルゴリズムを以下に示す．

**ディジタル署名**

DSA (FIPS 186-3)

**メッセージ認証(鍵付きハッシュ関数)**

HMAC，CMAC，CCM

**擬似乱数生成関数**

HMAC_DRBG (NIST SP 800-90),
Hash_DRBG (NIST SP 800-90),
CTR_DRBG (NIST SP 800-90),
ANSI X9.31 Appendix A.2.4 Using the 3-Key Triple DES (NIST-Recommended Random Number Generator Based on ANSI X9.31 Appendix A.2.4 Using the 3-Key Triple DES and AES Algorithms),
ANSI X9.31 Appendix A.2.4 Using AES (NIST-Recommended Random Number Generator Based on ANSI X9.31 Appendix A.2.4 Using 3-Key Triple DES and AES Algorithms),

**鍵共有**

MQV，ECMQV

次節以降で，各暗号に対する試験方法の概要を記述する．

### 2.3．試験項目作成の方針

試験項目は次の 2 つを基本方針として作成した．

- 暗号モジュール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．
- 多くのプラットフォーム上で実装された暗号モジュールを試験できるように，汎用性の高い試験方法を作成する．

まず暗号アルゴリズムごとの機能を抽出した．例えばブロック暗号の場合，暗号化，復号のように，暗号ライブラリ等で独立した API として提供されることが多いものを機能として抽出した．次に，機能ごとに，暗号アルゴリズムが仕様書通りに実装されていることを確認するために必要な試験項目を設定した．

### 2.4．ディジタル署名

**DSA (FIPS 186-3)**

試験対象機能は次の通り．

- 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
- 署名生成機能
- 署名検証機能

試験項目を以降に示す．

1. ドメインパラメータ生成機能
- IUT(試験対象モジュール)が生成した SEED および counter を JCATT に入力し，JCATT は FIPS 186-2 Appendix 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．
- $g^q \equiv 1 \bmod p$ であること．
- IUT が生成した複数のドメインパラメータ(p，q，g)が全て異なるものであること．

2. ドメインパラメータ検証機能
- JCATT が与えた前節に記述したドメインパラメータ生成機能に対する試験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 が与えた前節に記述したドメインパラメータ生成機能に対する試験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. 鍵ペア生成機能
- $y \equiv g^x \bmod p$ であること．
- $1 \leqq x \leqq q-1$，$2 \leqq y \leqq p-2$ であること．
- $y^q \equiv 1 \bmod p$ であること．
- IUT が生成した複数の鍵ペアが全て異なるものであること．

4. 署名生成機能
- JCATT が与えたプライベート鍵および平文に対して，IUT が生成した署名を，JCATT が署名検証した時に署名検証合格となること．
- 同じ平文，同じプライベート鍵に対して複数(別途規定する数) 署名を生成させた時，IUT が同じ署名を生成しないこと．

5. 署名検証機能
- JCATT が与えた正しい公開鍵，平文および署名，ならびに指定されたハッシュ関数に対して，IUT が正しく署名検

証合格と判定すること.

- JCATT が改竄した平文, 署名, または公開鍵に対して, IUT が署名検証不合格と判定すること.

## 2.5. メッセージ認証

**HMAC**

HMAC の試験項目を次に示す.

1. 短いメッセージに対する試験
2. 選択された長いメッセージに対する試験
3. 疑似ランダムメッセージに対する試験

**CMAC**

試験対象機能は次の通り.

- メッセージ認証子生成機能
- メッセージ認証子検証機能

試験項目を次に示す.

1. メッセージ認証子生成機能
- 短いメッセージに対する試験
- 選択された長いメッセージに対する試験
- 疑似ランダムメッセージに対する試験

2. メッセージ認証子検証機能
- JCATT が与えた正しい鍵, 平文およびメッセージ認証子に対して, IUT が検証合格と判定すること.
- JCATT が与えた改ざんされた鍵, 平文, またはメッセージ認証子に対して, IUT が検証不合格と判定すること.

**CCM**

試験対象機能は次の通り.

- 暗号化機能
- 復号機能

試験項目を次に示す.

1. 暗号化機能
- 種々の associated data に対する試験
- 種々の平文に対する試験
- 種々の nonce に対する試験
- 種々のメッセージ認証子に対する試験

2. 復号機能
- 与えられた暗号文, 鍵, メッセージ認証子長, nonce, associated data に対して, もとの平文に復号できること.
- 復号機能では INVALID 出力となるような, 暗号文, 鍵, メッセージ認証子長, nonce, associated data の組に対して, INVALID を出力すること.

## 2.6. 擬似乱数生成関数

擬似乱数生成関数の試験項目を次に示す.

- 種々のシードに対する試験
- モンテカルロ試験

## 2.7. 鍵共有

**MQV**

試験対象機能は次の通り.

- 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

試験項目を以降に示す.

1. ドメインパラメータ生成機能
- DSA と同様

2. ドメインパラメータ検証機能
- DSA と同様

3. 鍵ペア生成機能
- DSA と同様

4. 公開鍵検証機能
- 公開鍵 y およ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，そうでなければ不合格と判定すること．
  - $2 \leqq y \leqq p-2$ であること．
  - $y^q \equiv 1 \bmod p$ であること．

5. 鍵共有機能
- 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．

**ECMQV**

試験対象機能は次の通り．
- 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

試験項目を以降に示す．

1. ドメインパラメータ生成機能
- 生成されたドメインパラメータが正しい数学的関係を持つこと(仕様書に記載されている関係式をすべて満たすこと)．ECDSA は定義体標数によって試験項目が異なることに注意．
- 生成された複数のドメインパラメータが全て異なるものであること．

2. ドメインパラメータ検証機能
- DSA と同様

3. 鍵ペア生成機能

標数ｐの場合を以下に記述する．標数 2 の場合も同様である．
- $Q \neq O$ であること．
- $(Q_y)^2 \equiv (Q_x)^3 + a(Q_x) + b \bmod p$ であること．
- $nQ = O$ であること．
- $Q = dG$ であること．
- IUT が生成した複数の鍵ペアが全て異なるものであること．

4. 公開鍵検証機能
- DSA と同様

6. 鍵共有機能
- 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．

## 3. まとめ

JCATT は「暗号モジュールが“暗号アルゴリズム仕様書”に従って実装されているかどうかを検査する」ものである．実際の暗号モジュールの試験にあたっては，JCATT に実装された試験項目以外にも，適切なエラー処理がされているかどうかなど，暗号アルゴリズム仕様書には記述されていない様々な項目の試験が必要である．現時点の JCATT でカバーしていない試験項目をどう定め，どう試験するかなどの検討は今後の課題である．また，現時点の JCATT に含まれていない．暗号も存在する．それらの暗号に対する試験機能の追加も今後の課題である．

**参考文献**

[1] “CRYPTREC Report 2002 (暗号技術評価報告書),” IPA, TAO, 2003.

[2] L. E. Bassham III, “AESAVS,” NIST, 2002.

[3] L. E. Bassham III, “CCMVS,” NIST, 2004.

[4] L. E. Bassham III, “DSAVS,” NIST, 2004.

[5] L. E. Bassham III, “ECDSAVS,” NIST, 2004.

[6] L. E. Bassham III, “HMACVS,” NIST, 2004.

[7] L. E. Bassham III, “RNGVS,” NIST, 2005.

[8] L. E. Bassham III, “SHAVS,” NIST, 2004.

[9] S. S. Keller, “RSAVS,” NIST, 2004.

[10] NIST SP 800-17, “Modes of Operation Validation (MOVS): Requirements and Procedures,” S. Keller and M. Smid, NIST, 1998.

[11] NIST SP 800-20, “Modes of Operation Validation System for the Triple Data Encryption Algorithm (TMOVS) : Requirements and Procedures,” S. Keller, NIST, 2000.