社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2004－ＣＳＥＣ－26 (14)
2004／7／20

共通鍵暗号モジュールの試験に関する一考察

矢嶋 純† 武仲 正彦† 下山 武司‡

†‡富士通研究所 セキュアコンピューティング研究部

†〒674-8555 明石市大久保町西脇 64
‡〒211-8588 川崎市中原区上小田中 4-1-1

E-mail: †‡{jyajima, takenaka, shimo}@labs.fujitsu.com

あらまし テーブル参照実装にて実現した共通鍵暗号モジュールの試験について考察する。このようなモジュールの動作試験では、テーブル内の全データを参照する必要がある。本論文ではテーブルサイズと参照回数から全データ参照確率を導出する関係式を示し、NIST より公開されているモンテカルロ試験について全データ参照確率を検討する。そして実装によっては全データを参照しないことを実験によって示す。また全データ参照確率という観点から新しい動作試験を 2 手法提案する。

キーワード 共通鍵暗号、暗号モジュール、動作試験、モンテカルロ試験、NIST

On Validation Tests for Block Cipher Modules

Jun YAJIMA† Masahiko TAKENAKA† and Takeshi SHIMOYAMA‡

†‡FUJITSU LABORATORIES LTD, Secure Computing Laboratory

†64 Nishiwaki, Ohkubo-cho, Akashi, 674-8555, Japan
‡4-1-1 Kamikodanaka, Nakahara-ku, Kawasaki, 211-8588, Japan

E-mail: †‡{jyajima, takenaka, shimo}@labs.fujitsu.com

Abstract We consider validation tests for a block cipher module with table-lookup. In tests for these modules, all data in tables should be referred for validation. In this paper, we have examined a probability that all entries of tables are referred by test data. The probability is derived from the size of tables and amount of test data. Using the probability, we have considered the validation of tables about Monte Carlo tests published by NIST. As a result, we show that whether or not the NIST's Monte Carlo tests work depends on the implementation methods for the module. And we propose two revised validation testing methods.

Keyword Block Cipher, Block Cipher Module, Validation Test, Monte-Carlo test, NIST

1. はじめに

各種暗号のソフトウェア／ハードウェアモジュールを実装した場合、実装したアルゴリズムが仕様通りに実装されているかを試験する必要がある。一般的に暗号の場合、データの入出力空間が膨大であるため、平文／暗号文／秘密鍵としてとりうる全パターンを実際に入力して検証することはできない。このためモジュールの動作試験手法の確立は大きな課題となっている。NIST[1][2][3]や IPA[4]においても検討が進められており、NIST からは共通鍵暗号モジュールの評価手法として SP800-17, SP800-20, AESVS が公開されている。

共通鍵暗号を実装する場合、テーブル参照を行う手法を採用することが多い。テーブル参照実装での動作試験では、テーブル内の全データを最低 1 回参照し、平文／暗号文／秘密鍵の関係が正しいことを確認する必要がある。NIST の手法においても、特定の実装を用いた場合に全データを参照する試験が The Known Answer Test(KAT)として公開されている。一方、テーブル参照による実装では、高速処理や省サイズ化の目的に応じて実装者がテーブルのカスタマイズを行うことが多く、このようなモジュールでは KAT で全データを参照することは保証されない。また、同じく NIST の手法で公開されているモンテカルロ試験では、テーブル内の全データを参照するという観点からの検討は行われていない。IPA で行われている暗号モジュール検証ツールの検討においても、実装法依存の部分には

あまり深く立ち入らないことが報告書に記述されている。

そこで本論文ではモンテカルロ試験について、テーブル内の全データ参照確率という観点から検討を行う。本論文では、テーブルサイズと参照回数から全データ参照確率を導出する関係式について示し、NISTのモンテカルロ試験における全データ参照確率を導出する。また関係式の正当性について、実験を行って検討する。さらにテーブル内の全データを参照するという観点から動作試験を2手法提案する。

2. モジュールの動作試験

2.1. 暗号モジュールの動作試験

暗号モジュールとはハードウェア／ソフトウェアにて実現した暗号機能である。実装者が暗号モジュールを作成する場合、以下の2点に注意する必要がある。

1. 入力された平文に対し、暗号アルゴリズムの仕様に基づいて正しい暗号文を出力すること
2. ソフトウェア／ハードウェア特有のバグ（ソフト実装におけるメモリリークなど）がないこと

この内、本論文では共通鍵暗号について、1に関する検討を行う。1に関して、正しい動作を保証するには、平文／暗号文／秘密鍵について取りうる全ての組み合わせを実際に入力して検証する方法が考えられる。しかしこの検証方法では検証の手間が全数探索と同程度となるため、現実的な時間で試験を完了することができない。このため、NISTより公開されているSP800-17, SP800-20, AESVSにおいては、The Known Answer Testを用いて、S-boxや転置などの各パートについての正当性を検証し、それを積み重ねることでモジュール全体の正当性の検証を行っている。NISTより公開されている共通鍵モジュールの試験手法を表1にまとめる。これらの手法では、以下の3の手法が規定されている。

The Known Answer Test(KAT)

特定の実装法を用いて実現した暗号モジュールについて、S-boxテーブル、転置処理など、暗号化／復号／拡大鍵生成処理に含まれるパート毎の正当性を検証する試験である。試験は、特徴的な平文／暗号文／秘密鍵を使用して行う。

The Multi-block Message Test(MMT)

複数ブロックにまたがるデータについて、1ブロックの処理を繰り返し実行した場合と、複数ブロックの処理をまとめて実行した場合に出力されるデータを検証し、モジュールの動作が正しいことを検証する試験である。

The Modes Test(MCT)

ECB, CBC, OFBなどの各モードの正当性を検証する試験である。モンテカルロ試験を用いて、KATのみを合格するような設計にしていないかを検証する試験である。モンテカルロ試験では、数十万～数百万個の平文／暗号文／秘密鍵の関係を検証し、それらが正しければ合格となる。初期値（平文／秘密鍵）から最後に得られる出力（暗号文）までが依存する構造となっているため、途中に1回でも間違いがあると、極めて高確率で最終結果は誤った結果となる。

表 1. NISTの共通鍵暗号モジュール試験

試験手法	試験対象暗号	KAT	MMT	MCT
SP 800-17	DES[5]	有	無	有
SP 800-20	3DES[5]	有	無	有
AESVS	AES[6]	有	有	有

2.2. テーブル参照実装

共通鍵ブロック暗号の実装では、S-boxや転置などの処理をテーブル参照にて実現する方法がよく使用される。さらに、テーブル参照実装では、高速処理実装や省サイズ実装などの目的に応じてテーブルのカスタマイズを行うこともよく行われる。例えば、S-boxによる変換とそのあとに続く線形変換などの処理を事前に計算し、まとめて一つのテーブルで実現する方法(図1)や、隣り合うS-boxテーブル同士を一つにまとめる方法などがよく用いられる。(図 2)。

テーブル参照実装された共通鍵暗号モジュールの動作試験では、テーブル内の全データを最低1回参照し、その結果を検証する必要がある。これはテーブル内の各要素が独立であることから、k個の要素を持つテーブル参照処理がkヵ所に分岐する分岐処理と等価となるためである。分岐処理の動作試験では全分岐先を網羅する必要があるが、テーブル参照実装においてはこれがテーブル内の全データ参照にあたる。

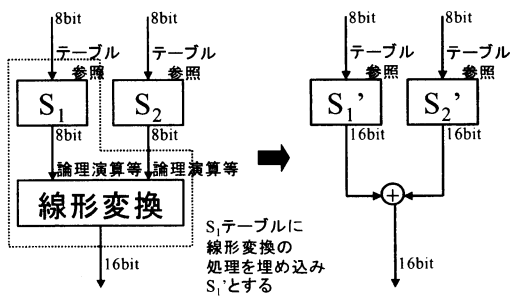


図 1. テーブルのカスタマイズ（1）

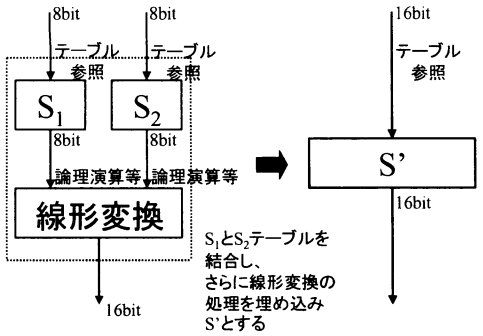


図 2. テーブルのカスタマイズ（2）

テーブル参照実装における試験合格条件：

暗号モジュール内の全てのテーブルについて、全データを最低1回参照し、そのときの平文／暗号文／秘密鍵の関係が仕様と等しいことを確認する。

§2.1で紹介した NIST の試験手法について、テーブル内の全データを参照するという観点から問題点を検討する。

The Known Answer Test

NIST の試験には、テーブル内の全データを参照するための平文／暗号文／秘密鍵が公開されているものがある。しかしこれらは特定の方法で実装された暗号モジュールについてのみ有効なデータである。従って、実装者が§2.2で示したようなカスタマイズを行って実装した暗号モジュールの場合には、全データを参照することは保証されない。特に図 2のように隣接するtビットの S-box テーブル同士を結合した場合、テーブルの要素数は 2^t 倍になるため、全データを参照する確率は大幅に低下すると考えられる。

The Multi-block Message Test

テーブル内のデータを参照することを意図して作成された試験ではなく、処理対象のデータ数も少ないため、テーブル内の全データを参照する試験としては適していない。

The Modes Test

テーブル内のデータを参照することを意図して作成された試験ではないが、モンテカルロ試験は処理回数が膨大であるため、テーブル内の全データを参照することが期待できる。しかし、実際どの程度の確率で全データを参照するかといった検討は行われておらず、処理回数に関する明確な基準も示されていない。

以上より、テーブル参照実装では、実装者によるテーブルのカスタマイズを考慮した場合、モンテカルロ試験(The Modes Test)を用いることでテーブル内の全データを参照することが期待できる。そこで本論文では、モンテカルロ試験における全データの参照確率と処理回数について検討を行う。

3. モンテカルロ試験

3.1. テーブル参照回数と全データ参照確率の関係

モンテカルロ試験において全データを参照する確率を導出する前に、まず参照値が k 通りのテーブルを n 回参照した場合にテーブル内の全データを参照する確率 $P_k^{(n)}$ を導出する。

[定理]

k 個の要素(s_1, …, s_k)からなるテーブルをランダムかつ独立に n (n≧k)回参照する場合に全データを参照する確率 $P_k^{(n)}$ は以下の式で表される。

$$p_k^{(n)} = \sum_{j=0}^{k-1} (-1)^j {}_kC_{k-j} \frac{(k-j)^n}{k^n} \qquad \cdots(1)$$

[証明]

テーブルを 1 回参照した際に、要素 s_i が参照される確率を a_i とする。テーブルをランダムに n 回索引した場合、$q_k^{(n)} = (a_1 + a_2 + \ldots + a_k)^n$ を展開した各項は、テーブル参照のパターンとその確率を表現する。例えば、s_1 が連続 n 回索引される確率は a_1^n で表現される。

ここで、a_i はテーブルの各要素を参照する確率なのですべて 1/k とする。また、n 回の参照で s_1～s_j の j 個(j≦k)のエントリすべてを参照する確率を $p_j^{(n)}$ とすると $q_k^{(n)}$ は以下の式で表現できる。

$$q_k^{(n)} = p_k^{(n)} + {}_kC_{k-1} \cdot p_{k-1}^{(n)} + {}_kC_{k-2} \cdot p_{k-2}^{(n)} + \cdots + {}_kC_1 \cdot p_1^{(n)}$$
$$= \sum_{i=0}^{k-1} {}_kC_{k-i} \cdot p_{k-i}^{(n)}$$

この関係式を用いて、(1) 式の右辺を $p_j^{(n)}$ で表現する。

$$\sum_{j=0}^{k-1} (-1)^j {}_kC_{k-j} \frac{(k-j)^n}{k^n} = \sum_{j=0}^{k-1} (-1)^j {}_kC_{k-j} \cdot q_{k-j}^{(n)}$$
$$= \sum_{i=0}^{k-1} \sum_{j=0}^{i} (-1)^j {}_kC_{k-j} {}_{k-j}C_{k-i} \cdot p_{k-i}^{(n)}$$
$$= \sum_{i=0}^{k-1} {}_kC_{k-i} \cdot p_{k-i}^{(n)} \sum_{j=0}^{i} (-1)^j {}_iC_j$$

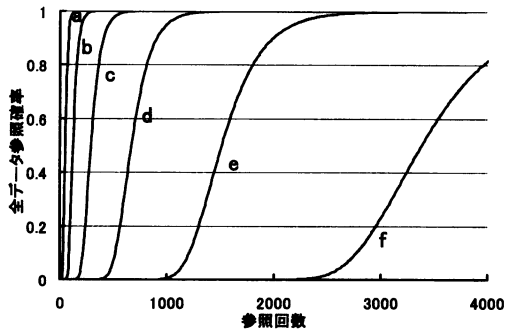
ここで、

$$\sum_{j=0}^{i} (-1)^j {}_iC_j = \begin{cases} 1 & (i=0) \\ 0 & (i \neq 0) \end{cases}$$

となることから、

$$\sum_{j=0}^{k-1}(-1)^j {}_kC_{k-j}\frac{(k-j)^n}{k^n}=p_k^{(n)} \qquad \square$$

4bit～9bit テーブルについて、(1)式を用いて全データ参照確率を導出した結果を図 3のグラフに示す。図3より、同じ参照回数でも、テーブルの大きさによって全データ参照確率は極端に異なっていることがわかる。



a：4bit テーブル，b：5bit テーブル，
c：6bit テーブル，d：7bit テーブル，
e：8bit テーブル，f：9bit テーブル

図 3. 全データ参照確率の推移

3.2. 暗号処理における全データ参照確率

実際の暗号処理における全データ参照確率について検討する。共通鍵暗号に関する典型的なモデルを図4に示す。図において、A～H はテーブル参照処理、L, op は線形変換などの処理を示す。

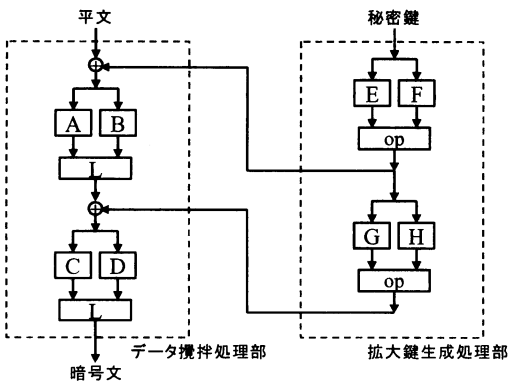


図 4. 典型的な共通鍵暗号のモデル

データ攪拌処理部（暗号化／復号処理）での全データ参照確率

i) 同一ラウンドのテーブルにおける検討

簡単のため、図 4の第 1 ラウンドのみから構成される暗号を例に検討する。同一ラウンドに並列に存在するテーブル A, B の参照値は、互いに依存関係が存在せず、独立の値となっている。また、入力値がランダムの場合、A, B の参照値もランダムとなる。このため(1)式を使用するための前提条件が成立している。よって、A, B が同一のテーブルの場合、テーブルの個数を参照回数としてカウントし、(1)式にて全データ参照確率を導出する。また、A, B が異なるテーブルの場合には、テーブル毎の全データ参照確率 P_{kA} , P_{kB} を(1)式で導出し、それらを掛け合わせた $P_{kA}\times P_{kB}$ が処理全体の全データ参照確率となる。

ii) 複数ラウンドのテーブルにおける検討

簡単のため、図 4の A, C のみからなる暗号について検討する。C の参照値は 1 ラウンド目の計算結果とそのラウンドの拡大鍵などに依存している。このため 1 ラウンド目の計算結果を導出するために使用した A の参照値にも依存しているといえる。しかし、一般的な暗号を使用する限り、拡大鍵の加算やその他の攪拌処理によって、ほぼ独立の振る舞いをすると考えられ、ランダムな値になっていると考えられる。そこで本論文では、異なるラウンドのテーブルについても(1)を利用するための前提条件が成立していると仮定する。この仮定によって A, C が同一のテーブルの場合、テーブルの個数を参照回数としてカウントし、(1)式にて全データ参照確率が導出できる。また、A, C が異なるテーブルの場合には、テーブル毎の全データ参照確率 P_{kA}, P_{kC}を(1)式で導出し、それらを掛け合わせた $P_{kA}\times P_{kC}$ が処理全体の全データ参照確率となる。

iii) 攪拌処理全体における検討

i), ii)での検討を攪拌処理全体に拡張する。本論文では、攪拌処理において、同一ラウンド内だけでなく異なるラウンドに存在するテーブルであっても、それらが参照する値は独立かつランダムであると仮定する。この仮定により、(1)式を利用するための前提条件が成立する。例えば、AES のリファレンス実装などのように、A～D が同一テーブルの場合、1 回の攪拌処理における参照回数 n=「テーブルの個数」として(1)式によって全データ参照確率を導出できると考えられる。また、DES の実装などのように、A と C、及び、B と D が同一テーブルの場合、テーブルの種類毎に参照回数 n_{AC} , n_{BD}をカウントし、テーブルの種類毎の全データ参照確率 $P_{kAC}^{(n_{AC})}$, $P_{kBD}^{(n_{BD})}$ を導出する。処理全体の全データ参照確率はそれらを掛け合わせた$P_{kAC}^{(n_{AC})}\times P_{kBD}^{(n_{BD})}$ となると考えられる。

拡大鍵生成処理部での全データ参照確率

拡大鍵生成処理についても E, F のような同一ラウンドのテーブルの参照値については i)の検討と同様に(1)式を使用する前提条件が成立していると考えられる。しかし、拡大鍵生成処理にはラウンド毎の拡大鍵の加

算等が存在せず、op の処理も単純なものが多いため、異なるラウンド間の参照値については(1)式を使用するための前提条件が成立していないことがある。例えば DES の PC2 などでは op が単純な処理のため、テーブル E と G, F と H には依存関係がある。このため、攪拌処理と同様の仮定を設定し、(1)式を用いて全データを参照する確率を導出することは、適切ではないと思われる。拡大鍵生成処理における全データ参照確率については今後検討の余地がある。

3.3. モンテカルロ試験における全データ参照確率

モンテカルロ試験のアルゴリズムの概要を以下に示す。モンテカルロ試験では、異なる u 個の秘密鍵ならびに v 個の平文を用いて、u 回の拡大鍵生成処理と u×v 回のデータ攪拌処理が（暗号化／復号処理）行われる。従って各テーブルの参照回数は、「拡大鍵生成処理 1 回における独立な参照回数×u」、及び、「データ攪拌処理 1 回における独立な参照回数×(u×v)」となり、(1)式による全データ参照確率計算の際にこの値を使用することで、試験全体における全データ参照確率が導出できる。

モンテカルロ試験のアルゴリズム（ＥＣＢモード）

Initialize KEY_0, P_0;

for i=0 to u {
    Record i, KEY_i, P_0;
    for j=0 to v {
        C_j=CRYPTO(KEY_i, P_j);
        P_{j+1}=C_j;
    }
    Record C_j;
    KEY_{i+1}=$KEY_i \oplus C_j$;
    P_0=C_j
}

3.3.1. NIST の試験における問題点

NIST の動作試験におけるモンテカルロ試験では、暗号化／復号処理に比べて秘密鍵の更新回数が少ない。このため拡大鍵生成処理で数回しか参照しないテーブルについては、全データを参照する確率が低い可能性がある。このことを以下の実験で検証する。

実験

DES、3DES の PC1 をテーブル参照実装したモジュールについてモンテカルロ試験を行った場合の全データ参照確率を実際に実装して検証する。DES 及び 3DES の PC1 処理は論理演算にて実現することが多いが、今回の実験ではこの処理をテーブル参照にて実現した場合について検討する。PC1 処理では 8bit おきに存在するパリティを考慮すると 7bit テーブルを使用するのが効率的と考えられる。しかし先頭から 8bit ずつ取り出し、パリティを除いた 7bit を参照値として使用するような実装を行った場合、各テーブルは 7bit 入力 56bit 出力となり、一部のプラットフォームを除いて非効率となる。そこで図 5、図 6のように(4bit+3bit)、あるいは(3bit+4bit)を取り出して参照値とする実装を行った。これにより各テーブルは 7bit 入力 28bit 出力となり、多くのプラットフォームで効率的となることから、検討対象として本実装を使用することにした。本実装を適用した DES モジュールに SP800-17、3DES モジュールに SP800-20 のモンテカルロ試験を適用した際の全データ参照確率を検討する。

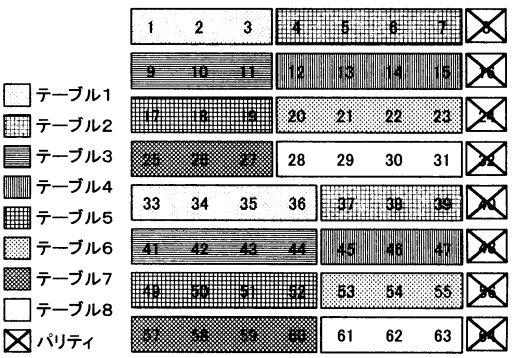


図 5. PC1 のテーブル参照実装

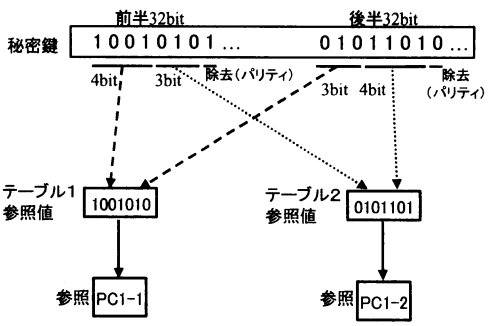


図 6. PC1 のテーブル参照値作成処理

試験方法

DES、2key-3DES-ECB、3key-3DES-ECB について、SP800-17、SP800-20 と同じループ回数のモンテカルロ

試験を 1,0000 回実行し、全データ参照確率を導出する。ただし初期秘密鍵＝乱数とする

結果

暗号化／復号について、それぞれ表 2の結果が得られ、理論値とほぼ一致していることが判明した。理論値の導出方法は以下に記述する。

表 2. NIST の動作試験における全データ参照確率

処理	全データ参照数／総数	全データ参照確率実験値(理論値)
DES-ECB 暗号化	0 / 10000	0% (4.40×10^{-20}%)
DES-ECB 復号	0 / 10000	0% (4.40×10^{-20}%)
2key-3DES-ECB 暗号化	1511 / 10000	15.11% (14.3%)
2key-3DES-ECB 復号	1425 / 10000	14.25% (14.3%)
3key-3DES-ECB 暗号化	9215 / 10000	92.15% (92.3%)
3key-3DES-ECB 復号	9214 / 10000	92.14% (92.3%)

理論値

PC1 処理は拡大鍵生成処理で 1 回しか行われず、また入力鍵に依存する構造のため、各参照値は独立と考えて問題ない。このため、(1)式にて全データ参照確率を導出することができる。

DES における全データ参照確率の理論値

SP800-17 のモンテカルロ試験では秘密鍵を 400 回更新する。また PC1 テーブル 8 個は拡大鍵生成処理でそれぞれ 1 回ずつ参照される。このため、各テーブルは試験全体でそれぞれ 400 回参照され、全データを参照する確率を(1)式で導出すると 0.214%となる。従って、全体では $0.00214^8 \fallingdotseq 4.40\times10^{-22}$ となり、全データを参照する確率は極めて低いことが理論的に導かれる。

3DES における全データ参照確率の理論値

SP800-20 のモンテカルロ試験では秘密鍵を 400 回更新する。また PC1 テーブル 8 個は拡大鍵生成処理で、2key-3DES の場合に各 2 回、3key-3DES の場合に各 3 回参照される。このため、2key-3DES の場合、各テーブルは試験全体で 800 回参照され、全データを参照する確率を(1)式で導出すると 78.4%となる。従って、全体では $0.784^8 \fallingdotseq 0.143$ となる。また、3key-3DES の場合、各テーブルは試験全体で 1200 回参照され、全データを参照する確率を(1)式で導出すると 99.0%となる。従って、全体では $0.99^8 \fallingdotseq 0.923$=92.3%となる。

NIST の初期値の場合

SP800-20 用に、モンテカルロ試験のサンプル初期値が公開[7]されている。このサンプル値を今回の実験で使用した DES、3DES モジュールに用いた場合について検証を行った。結果を表 3に示す。なお、参照されていない箇所に故意に誤ったデータを挿入して試験を実施したところ、挿入前と同じ結果が得られた。これにより、実装の誤りを検出できない場合があることが実際に確かめられた。

表 3. NIST の初期値におけるデータ参照実験

処理	全データが参照されたテーブル数	参照された要素数
DES-ECB 暗号化	0 / 8	968 / 1024
DES-ECB 復号	0 / 8	979 / 1024
2key-3DES-ECB 暗号化	7 / 8	1023 / 1024
2key-3DES-ECB 復号	6 / 8	1022 / 1024
3key-3DES-ECB 暗号化	8 / 8	1024 / 1024
3key-3DES-ECB 復号	7 / 8	1023 / 1024

3.4. 新しい試験の提案

§3.3.1の実験より、NIST の動作試験におけるモンテカルロ試験では、実装によっては全データが参照されないテーブルが存在することが確認された。本節では、全データの参照確率の観点から以下の 2 つの手法を提案する。

提案 1．The Known Answer Test の拡張

実装者がデータを作成する試験方法である。試験方法、及びメリット、デメリットは以下のとおりである。

試験方法

1．実装者が、自分の実装したモジュール内の全テーブルについて、テーブル内の全データを参照するデータを自ら作成する。
2．作成したデータが正しいことをリファレンスコードにて確認する。

メリット

・ テーブル内の全データを 100％の確率で参照する。

デメリット

・ 実装者自らが生成するため手間がかかる。
・ データ作成ミスが発生する可能性がある。
・ 第 3 者による客観的評価が不能である。

なお、本試験を行うためには、認定機関などが公式なリファレンスコードを公開することが前提となる。

提案 2．モンテカルロ試験の拡張

「テーブルエントリ数の最大値」と「目標とする全データ参照確率」をあらかじめ規定し、そのサイズま

でのテーブルについて目標とする確率で全データを参照することを保証した試験である。試験方法、及び、メリットとデメリットは以下のとおりである。

1．全データを参照する確率の目標値 $p_k^{(n)}$と、目標とする全データ参照確率を保証する処理（拡大鍵生成処理／暗号化処理／復号処理）を決定する。
2．1で決定した処理について、$p_k^{(n)}$の確率を保証する最大テーブルエントリ数 k を決定する。
3．2で決定した k について、$p_k^{(n)}$を達成するための実行回数 n を導出する。
4．1で決定した拡大鍵生成処理／暗号化処理／復号処理について、1回の処理につき1回のテーブル参照が行われるという前提で、モンテカルロ試験での外部ループ数 u を決定する。つまり、拡大鍵生成処理であれば u=n、暗号化／復号処理であれば u=n/v としてモンテカルロ試験を実行する。

メリット

・ 第3者による客観的評価が可能である。
・ 実装者がデータを生成しないためミスが少ない。

デメリット

・ 全データの参照が確率的な保証となる。

なお、本試験を行うためには、認定機関などが、モンテカルロ試験の仕様に基づいた大量の公式なリファレンスデータを公開することが前提となる。データは、モンテカルロ試験の内部ループ数 v を固定し、外部ループ数 u を十分な数として大量に生成しておく。

4. まとめ

本論文では共通鍵暗号のテーブル参照実装の動作試験について検討を行った。テーブル参照実装の動作試験では、テーブル内の全データを最低1回参照することが必要となる。本論文ではテーブルサイズと参照回数を用いて、全データ参照確率を導出する関係式を示した。NIST のモンテカルロ試験について実験を行ったところ、この関係式で導いた全データ参照確率と実験結果がほぼ等しくなることを確認した。また本論文では、NIST のモンテカルロ試験について、実装によっては全データを参照しないことを示し、全データ参照確率という観点から新しい動作試験について2つの手法を提案した。この手法によりカスタマイズしたテーブルについても全データの参照を高確率で保証できる。今後の課題としては、拡大鍵生成処理において、テーブル参照値に従属関係が存在した場合についての全データ参照確率の検討、及び公開鍵暗号の動作試験についての検討が考えられる。

文　献

[1] NIST Special Publication 800-17, "Modes of Operation Validation System(MOVS): Requirements and Procedures," Sharon Keller and Miles Smid. U.S. Department of Commerce / National Institute of Standards and Technology, February 1998.

[2] NIST Special Publication 800-20, "Modes of Operation Validation System for the Triple Data Encryption Algorithm(TMOVS): Requirements and Procedures," Sharon S.Keller. U.S. Department of Commerce / National Institute of Standards and Technology, April 2000.

[3] "The Advanced Encryption Standard Algorithm Validation Suite(AESVS)," Lawrence E.Bassham III. National Institute of Standards and Technology, November 2002.

[4] "電子政府行政情報化事業　暗号化モジュール検証ツールのための技術仕様の開発　調査報告書," 情報処理推進機構, February 2004.

[5] FIPS PUB 46-3, "DATA ENCRYPTION STANDARD(DES)," U.S. DEPARTMENT OF COMMERCE / National Institute of Standards and Technology, October 1999.

[6] FIPS PUB 197, "ADVANCED ENCRYPTION STANDARD(AES)," U.S. Department of Commerce / National Institute of Standards and Technology, November 2001.

[7] Advanced Encryption Standard (AES), Data Encryption Standard (DES), Triple-DES, and Skipjack Algorithms "Triple-DES Sample Vectors," http://csrc.nist.gov/cryptval/des/tripledes-vectors.zip