Mutable S-box に対する安全性評価

鎌田 真吾　山内 志保　長瀬 智行

弘前大学理工学部　〒036-8224　青森県弘前市大字文京町 3
E-mail:　nagase@eit.hirosaki-u.ac.jp

あらまし　本研究室では，ユーザが使用する暗号鍵によって出力値が変化するように発展させた Mutable S-box を提案した．しかし，その安全性評価は S-box 単体での差分／線形解読法に対する評価に留まっている．そこで本研究では，Mutable S-box をプログラム上で AES に実装する手法を提案した．そして，暗号における安全性評価の基本となる乱数性の評価を行った．

Security Level Evaluation of a Mutable S-box

Shingo Kamata　Shiho Yamauchi　Tomoyuki Nagase

Faculty of Science and Technology, Hirosaki University
3 Bunkyo-cho, Hirosaki-shi,Aomori, 036-8224 Japan
E-mail:　nagase@eit.hirosaki-u.ac.jp

Abstract　This report scrutinizes a mutable S-box of AES, which has been proposed in for improving the complexity of the S-Box's structure, looking for vulnerability to differential cryptanalysis and especially the linear cryptanalysis. The structure of the AES S-box has been expanded and modified to be congruent with the proposed algorithm and to obtain appropriate non-linearity of the S-box. The Cryptanalysis of mutable S-box is based on a statistical test for randomness to measure the unpredictability level of the output values.

１．序論

差分／線形解読法の出現により DES**[1]**の安全性が崩壊したため，1997 年，米国商務省標準技術局（NIST : National Institure of Standard and Technology）は DES に代わる新たな次世代標準暗号方式 AES**[2]**のアルゴリズムを公募し, 2001 年, SPN 構造を持つ Rijndael**[3]**が採用された．以降, AES は米国のみならず，欧州や日本の暗号規格にも採用され，近年においても，共通鍵暗号方式の代表的な規格として幅広く利用されている．

しかし，AES におけるアルゴリズム内の非線形変換処理は, DES と同様に図 1 に示すような S-box (Substitution-box)による換字処理に依存している．現在, AES に対する攻撃対象の約 80%がこの換字処理部(S-box)となっており，換字処理部を改善することで安全性の向上が見込まれる**[4]**.

そこで本研究室では，ユーザが使用する暗号鍵によって出力値が変化するように発展させた Mutable_S-box（以下 M_S-box）を提案した**[4]**．これによって，差分解読法，線形解読法における安全性の向上が図れる. しかし, その安全性評価は, S-box 単体での差分／線形確率の比較のみであり, 実用化のための評価が不十分である．よって，本研究では M_S-box をプログラム上で AES に実装する手法を提案した．そして，通常の AES と M_S-box を実装した AES（以下 AES-MS）で暗号化された値に対して，安全性評価の基本となる乱

数性の評価を行い，その結果を比較した.

また，実装したことによって，処理時間にどの程度の違いが発生したのかの比較も行った.

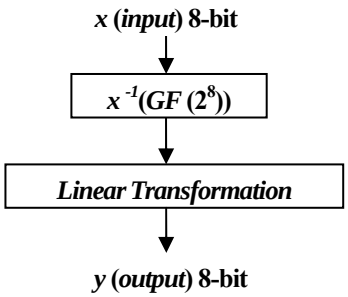

図 1 : AES で用いられる従来 S-box の構成

2. M_S-box について

2.1. M_S-box の構成

M_S-box の構成は下図に示すように, 従来 S-box の構成に Splitting Process，ガロア体 GF(2^5), GF(2^3) 上での逆元演算，Combining Process の処理を加えた構成となっている.

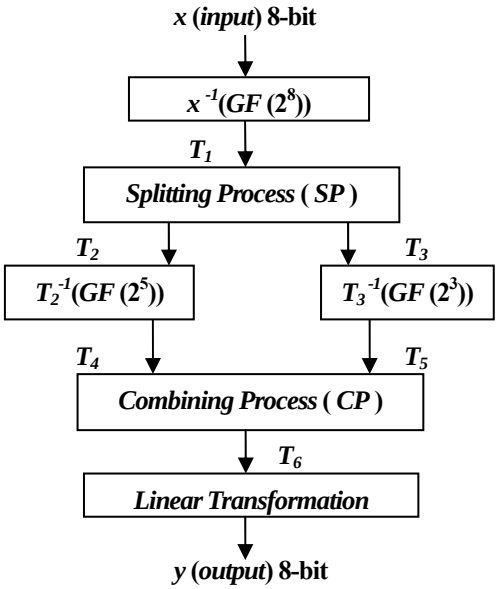

図 2 : Mutable_S-box の構成

M_S-box では初めに, 入力データ x (8-bit)に対して，ガロア体 GF(2^8)上での乗算の逆元を算出しこれを T_1 とする．次に，Splitting Process (*SP*) にて並び替え処理を行い，T_2 (5-bit)および T_3 (3-bit)に分割する．分割されたデータ T_2,T_3 に対して，それぞれガロア体 GF(2^5)上，GF(2^3)上での乗算の逆元 T_4 (5-bit)と T_5 (3-bit)を求める．そして，Combining Process(*CP*) にて T_4, T_5 の結合を行い T_6 (8-bit)とし，そのデータに対して線形変換処理を行い，演算結果 y (8-bit)を M_S-box の出力データとする．ここで，SP において 8-bit データを 5-bit データおよび 3-bit データに分割しているのは，ガロア体の位数の次数が偶数のときより，奇数のときの方が，差分／線形解読法に対する安全性を有することが予想されているからである**[5]**．予想されている安全評価指標の期待値を以下に示す.

表 1 : ガロア体上での安全性評価指数

	MDP	MLP
GF(2^3)上でのべき乗演算	2^{-2}	2^{-2}
GF(2^4)上でのべき乗演算	2^{-2}	2^{-2}
GF(2^5)上でのべき乗演算	2^{-4}	2^{-4}
GF(2^6)上でのべき乗演算	2^{-4}	2^{-4}
GF(2^7)上でのべき乗演算	2^{-6}	2^{-6}
GF(2^8)上でのべき乗演算	2^{-6}	2^{-6}

2.2. Splitting Process（SP）

M_S-box の入出力パターンが暗号鍵によって変化するようにしている処理部が *SP* であり，M_S-box の中心となる処理部である.

SP では初めに，暗号鍵より生成されるサブ鍵(15-bit)を用いて，2 つのパラメータ α と N を算出する．この，α と N を用いて並び替え処理を行っている．ここで，α を Initial point（並び替え開始位置），N を Kernel value（ビットナンバー選択値）とする.

サブ鍵より 2 つのパラメータ α と N を算出する手順は，まず 15-bit データであるサブ鍵を上位から α（3-bit），β（6-bit），γ（6-bit）に分割する．この時点で α の値は決定される．そして，β の各 bit の排他的論理和を算出し，その結果が 1 であれば β の上位 3-bit, 0 であれば β の下位 3-bit を X とする．ここで，X が {0, 0, 0} である場合，(R mod

6)+2 の演算結果を X に上書きしている．(R はラウンド数)．γ に関しても同様の処理を行い，結果を Y とする．この，X と Y を元にして，ガロア体 GF(2^3)上での乗算を行い，演算結果を N とする．

ここで，N が {0, 0, 1} である場合，(R mod 6)+2 の演算結果をNに上書きしている．これによって，Nの値が $2 \leqq N < 8$ の範囲内に収まるようにしたのは，並び替え位置を GF(2^3) 上のべき乗演算 $N^m(0<m<8)$によって算出しているためである．もし，N の値が 0 や 1 となると，m にかかわらず値が固定されてしまい，並び替え処理が正常に行われないからである．

上記の手法で算出されたαとNを元に並び替え処理を行っている．具体的には，並び替え後のデータを格納する配列に P[0] から P[6] までラベル付けを行い，$P[\alpha]$ に b_0 を，$P[(\alpha+m) \bmod 8]$ に $b_N{}^m$ を格納することによって，並び替えを行っている．処理例を表 2 に示す．

表 2 : SP での処理例（α=3，N=2 の場合）

Label (並び替え後の配列)	Compute of Bit number	Bit number
P[0]	$N^5=2^5=7$	b_7
P[1]	$N^6=2^6=5$	b_5
P[2]	$N^7=2^7=1$	b_1
P[3]	Initial Point	b_0
P[4]	$N^1=2^1=2$	b_2
P[5]	$N^2=2^2=4$	b_4
P[6]	$N^3=2^3=3$	b_3
P[7]	$N^4=2^4=6$	b_6

2.3. M_S-box の安全性評価

M_S-box では安全性評価の基準として最大平均差分確率 (MADP), 最大平均線形確率 (MALHP), Highest MDP (HMDP), Highest MLP (HMLP)の 4 つの指標を示している**[4]**.

ここで，HMDP と HMLP は当研究室で考案された評価方法で，可変 S-box に使用される指標である．これは，α と N の値によって生成される 48 個（α が 8 通り，N が 6 通り）の入出力パターンそれぞれにおいて，安全性が最も低い入出力パターンの指数を表わしたものである．従来 S-box と M_S-box を比較した結果を表 3 に示す．これらの安全性評価指数は文献**[6]**の評価ライブラリ関数を使用して算出している．

表 3 : 従来 S-box と M_S-box の安全性評価

	MADP	MALHP	HMDP	HMLP
AES S-box	2^{-6}	2^{-6}	2^{-6}	2^{-6}
M_S-box	$2^{-6.42}$	$2^{-6.67}$	$2^{-4.19}$	$2^{-3.36}$

表の数字は各種解読法の成立する確率を示しており，確率が低ければ低いほど安全性が高いとされている．表 3 より MADP／MALHP に関しては従来 S-box よりも安全性が向上していることがわかる．一方 HMDP／HMLP に関しては従来 S-box よりも安全性が低下している．これは，M_S-box で生成される 48 個の入出力パターンを個々に見ていった場合に，従来 S-box よりも MDP／MLP において安全性が低いパターンが存在していることを意味している．つまり，個々でみると従来 S-box よりも劣っているが，48 通りのパターンがあるため，全体でみると安全性は向上しているのである．しかし，個々の安全性が低いパターンを狙われる危険があるため，HMDP 及び HMLP においても従来 S-box の安全性を保持することは必須条件である. これは, 今後の大きな課題である.

3. 標準暗号方式 AES への実装

3.1. AES へのプログラム上での実装

これまで，M_S-box について仕様や評価をまとめてきたが，これらは全て M_S-box 単体での評価にとどまっている．M_S-box の最終目標は標準暗号方式 AES への実装であるため, 実際に実装した後の評価に関してもデータ収集しなければならない．そこで，本研究では初めに，プログラム上での AES への実装方法を提案する．

3.2. 実装方法

今回は AES のプログラムに対して，SubBytes 内の処理を M_S-box に変更することで，プログラム上での実装を行った．サブ鍵の決定方法に関しては，ラウンド鍵を用いて図 3 のように行うこととする．

初めに，ラウンド鍵 128-bit を 2 つ用意し，それぞれ 8-bit ずつに分割し K_i (i は 0～15) とする．次に，1 つは K_i 以外の XOR を，もう一方は $K_{(i+1)}$ 以外の XOR を計算しそれぞれ出力する．ここで，$K_{(i+1)}$ の方は，最下位 bit を除去する．最後に，2 つの出力 8-bit と 7-bit を結合し，サブ鍵(15-bit)としている．AES のブロック長は 128-bit，S-box の入出力は 8-bit より 1 つのブロックでは M_S-box を 16 個使用することになるが，図 3 の方法だと，その全てにおいて異なるサブ鍵を用いることが出来る．

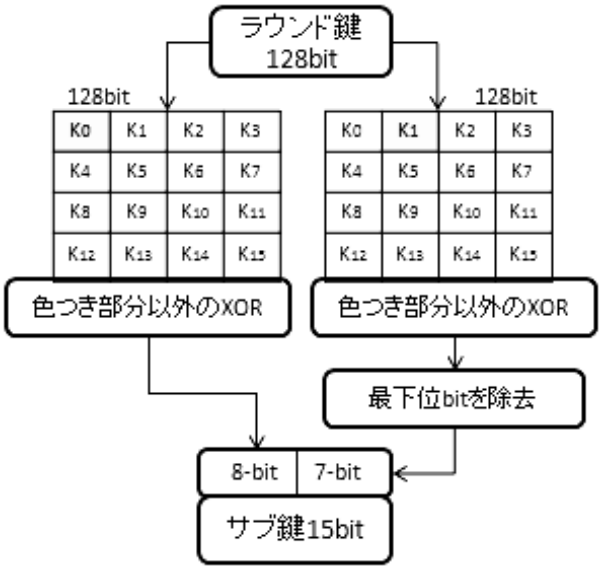

図 3 : サブ鍵の決定方法（i=0 の場合）

4. 実装後の評価

4.1. 処理時間の比較

実装後，通常の AES と AES-MS のプログラムを比べると非常に大きな違いが存在する．それは，AES の S-box は換字表を用いて SubBytes 変換が出来るため，SubBytes 変換でのプログラムが単純であり，速度の向上がみられる．一方，M_S-box では，サブ鍵によって何通りもの入出力パターンが存在するため，換字表をあらかじめ作成することは難しく，大きな遅延が予想される．しかしその分，入力から出力を推測することは困難であるため安全性の向上がみられる．そこで，AES と AES-MS では，処理速度にどの程度の違いがあるのか，プログラム上で計測した．表 4 は S-box 単体での比較で，表 5 が実装後の鍵長別の比較，表 6 が 3 つの鍵長での暗号化を 1 回としたときの繰り返し回数による比較である．表の結果は，全てにおいて 1 万回分の平均を取ったものである．

表 4 : 従来 S-box と M_S-box の処理時間の比較（sec）

	従来 S-box	M_S-box
S-box 単体での処理時間	0.000561	0.000843

表 5 : 鍵長別の処理時間の比較（sec）

	AES	AES-MS
128-bit の処理時間	0.00117	0.00221
192-bit の処理時間	0.00127	0.00237
256-bit の処理時間	0.00133	0.00261

表 6 : 回数による処理時間の比較 (sec)

	AES	AES-MS
1 回分の処理時間	0.00149	0.00446
10 回分の処理時間	0.01319	0.04285
100 回分の処理時間	0.13411	0.44129

結果をみてみると，S-box 単体ではそれほど差がないことがわかる．しかし，実装してみると約 2 倍，処理数が多くなると約 3 倍の差となっていることがわかった．もちろん，コーディングによるものもあるかもしれないが，実装することによって M_S-box 全て（16 個×ラウンド数）でラウンド鍵からサブ鍵を求める処理を行ったり，換字表を用いていないため，その分遅延があるのは当然である．しかし，2～3 倍という値は決して小さな値ではないので，今後処理の高速化に関しても十分に考慮しながら研究を進めていく必要がある．

4.2. 乱数検定による安全性評価

暗号における乱数性というものは必須条件であり，乱数性がないものはその偏りから簡単に解読されてしまう危険性がある. AES に M_S-box を実装したことで，暗号化後の暗号文に対する乱数性に何らかの偏りが生じたのならば，根本から M_S-box の処理を見直さなければならないことになる．そのため，M_S-box に対する安全性評価の第一歩として乱数性の安全を証明する必要がある.

4.2.1. NIST 乱数検定

今回乱数検定に用いたのが，NIST Special Publication 800-22**[7]**である．この乱数検定は，表 7 のような 15 種類の検定法で構成されている.

表 7 : NIST 乱数検定に含まれる検定法

	検定名		検定名
1	一次元度数検定	9	重なりあいのあるテンプレート検定
2	ブロック単位の頻度検定	10	Maurer のユニバーサル統計検定
3	累積和検定	11	近似エントロピー検定
4	連の検定	12	ランダム偏差検定
5	ブロック単位の最長列検定	13	種々のランダム偏差検定
6	2 値行列ランク検定	14	系列検定
7	離散フーリエ変換検定	15	線形複雑度検定
8	重なりあいのないテンプレート検定		

乱数検定では，0 と 1 の乱数列を対象に，各検定法によって，*p*-value という値が算出される．この値を用いた判定基準**[7]**を以下に示す.

(1) *p*-value が 0.01 以上になる割合（*P*）
(2) *p*-value の一様性（*U*）

(1) は，乱数列の個数を *m* としたとき，*p*-value が 0.01 以上になる割合が (式 1) の範囲にある場合，良い擬似乱数生成器であると判定される.

$$0.99 \pm 3\sqrt{\frac{0.99 \times 0.01}{m}} \qquad (式1)$$

(2) は，区間[0,1)を 10 分割し，各区間に属する *p*-value の個数が均等であるかどうかを x^2 分布によって検定する．具体的には，$1 \leqq i \leqq 10$ について，F_i を区間[(*i*-1)/10,*i*/10)に属する *p*-value の個数とする時，(式 2) を計算し，p-value=igamc(9/2,x^2/2) を計算する．p-value≧0.0001 のとき，良い擬似乱数生成器であると判定される.

$$\chi^2 = \sum_{i=1}^{10} \frac{(F_i - m/10)^2}{m/10} \qquad (式2)$$

これらの判定を 500 本～1000 本行い，良い乱数生成器と判定された割合をもとに，合否を決めている.

4.2.2. 乱数性の測定結果

今回は, AES と AES-MS の 2 つについて乱数検定を行い, 結果を比較した. AES には鍵長 128bit, 192bit，256bit の 3 パターンがあるが，全てにおいて乱数性を調べるために，出力値を AES128，AES192，AES256，AES128，AES192 … のように交互に並べ検定を行った．また，入力値 128-bit と鍵 128-bit，192-bit，256-bit はあらかじめランダムに 300 万個ずつ用意しそれを使用した.

表 8 : AES に対する乱数検定結果

	A		B		C		D	
	U	P	U	P	U	P	U	P
1	○	○	○	○	○	×	×	○
2	○	○	×	○	×	○	×	○
3	○	○	○	○	×	×	×	○
4	○	○	×	○	×	○	×	○
5	○	○	○	○	○	○	×	○
6	○	○	○	○	○	○	○	○
7	○	○	○	○	○	○	○	○
8	×	×	×	×	×	×	×	×
9	○	○	×	○	×	○	○	○
10	-	-	○	○	○	○	○	×
11	○	×	×	○	×	×	×	○
12	○	○	○	○	○	○	○	○
13	○	○	○	○	○	○	○	○
14	○	○	×	○	×	×	×	×
15	○	○	○	○	○	○	○	○

U: p-value の一様性　　P: p-value が 0.01 以上になる割合
○: 合格　　×: 不合格　　ー: 未検定
A:10 万ビット×1000 本　B:100 万ビット×500 本
C:100 万ビット×1000 本　D: 100 万ビット×500 本(入力 0)

さらに，入力値が全て 0（偏りの大きい平分を想定）の場合の出力値についても乱数性を調べて

みた．なお，AES と AES-MS には全く同じ平分及び鍵を用いて比較した. 結果を表 8, 表 9 に示す.

結果を見てみると，AES，AES-MS 共に 10 万 bit の場合には，ほぼ合格しているが，100 万 bit になると約半数が不合格となっていることがわかる. しかし, 全ての値を細かく見ていったところ，ほとんど偏りはなく，あと一歩のところで不合格となっていた．これは，検定に関する問題で，仮に真のランダム性を持った乱数列を NIST 乱数検定に通した場合でも，全ての検定項目において合格する確率は，二項分布を用いて概算すると約 54%，正規分布を用いた場合は，約 78%となることがわかっている**[8]**．つまり，すべての検定に合格したからといって良い乱数性を持つとはいえないのである.

表 9: AES-MS に対する乱数検定結果

	A		B		C		D	
	U	P	U	P	U	P	U	P
1	○	○	○	○	○	○	○	○
2	○	○	×	○	×	×	×	×
3	○	○	×	○	×	○	×	○
4	○	○	×	○	×	×	○	○
5	○	○	×	○	×	○	○	○
6	○	○	○	○	○	○	○	○
7	○	○	○	×	×	×	○	○
8	×	×	×	×	×	×	×	×
9	○	○	○	○	○	○	×	○
10	-	-	○	○	○	○	○	○
11	×	×	×	○	×	×	×	○
12	○	○	○	○	○	○	○	○
13	○	○	○	○	○	○	○	○
14	○	○	×	○	×	×	×	×
15	○	○	○	○	○	○	○	○

※文字の定義に関しては表８と同様

しかし, 今回は AES と AES-MS の比較であり，細かく値を調べても大きな偏りや違いはなかった. よって，AES に M_S-box を実装したことで，乱数性に何らかの偏りが発生することはないといえる. また，入力が全て 0 の場合（かなりの偏りがある平分）でも，AES や AES-MS を通すことによって偏りのない暗号文が得られることも確認できた.

5. まとめ

今回, M_S-box をプログラム上で AES に実装し, その乱数性を調べることで, AES-MS は AES と同等の乱数性を持つことが証明された．これによって，M_S-box に対する安全性評価の第一歩を踏み出した．そして，今後の課題としては以下のことが挙げられる.

- HMDP／HMLP の更なる安全性の確保
- 実装後の処理速度の高速化
- 乱数性以外の安全性評価

HMDP／HMLP に関しては, 原因を解明しつつ, 可変にしても安全性が低下しないという方法を考えていかなければならない. 処理速度に関しては, 従来よりも遅くなるのは当たり前である. しかし, 近年のコンピュータ処理能力では，ほとんど変わらないということもありえるので，最低限度の速度を確立し，それを維持出来れば問題はない.

これらのことを踏まえた上で，M_S-box のアルゴリズムを改良してみる必要がある．そして実装後の安全性評価として, SQUARE 攻撃や高階差分／線形解読法，最大期待差分／線形確率(MEDP／MELP)などの評価も随時行っていく必要がある.

文 献

[1] “Data encryption standard (DES),” FIPS PUB 46-2, Dec. 30, 1993.

[2] Announcing the ADVANCED ENCRYPTION STANDARD (AES),”FIPS PUB 197, Nov. 2001.

[3] Joan Daemen, Vincent Rijmen, “The Rijndael Block Cipher,” Sep. 1999.

[4] A. Watanabe, et al., “A New Mutable Nonlinear Transformation Algorithm for S-box,” IEEE FINA 07, Vol.1, pp. 246-251, 2007.

[5] 松井 充, “ブロック暗号アルゴリズム MISTY,” ISEC96-11, 1996.

[6] 金子 敏信, “共通鍵ブロック暗号のための強度評価ライブラリ,” 東京理科大学理工学部電気工学科 金子研究室, Feb. 2004

[7] Andrew Rukhin, et al., “A Atatistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications SP800-22 rev1a,” April. 2010.

[8] U.S. Department of Commerce, National Institute of Standards and Technology, “Secure Hash Standards,” Aug.2002.