暗号 AES の SubByte 変換の評価

加藤知美 (指導教員：萩田真理子)

1 研究背景

DES は 1976 年に標準暗号として採用され, ブロック暗号の代表的な規格として広く使われてきた. しかし 1997 年アメリカ合衆国において, 急速な計算機の処理速度の上昇によって安全ではなくなったことを理由に, 新しい規格として AES(Advanced Encryption Standard) が世界規模で公募された. そして 2001 年 3 月,J.Daemen と V.Rijmen が提案した Rijndael が AES として採択され公表された. 以降,AES はアメリカ合衆国のみだけでなく, 欧州の暗号規格 NESSIE や日本の暗号規格 CRYPTREC にも採用され, 今日の共有鍵暗号方式の代表的な規格の一つとして利用されており, 新たな共有鍵暗号方式の規格を提案する際の速度や安全性の評価基準としても利用されている.

本研究では,AES の変換のひとつである SubByte 変換における乱数性と AES の変換を一部を抜いたときの乱数性を評価した.

2 AES(Advanced Encryption Standard)

2.1 仕様

ブロック長 128bit,192bit,256bit の中から選択可能
鍵長 128bit,192bit,256bit の中から選択可能
ラウンド数 鍵長に依存. 鍵長 128bit:10 回,192bit:12 回,256bit:14 回

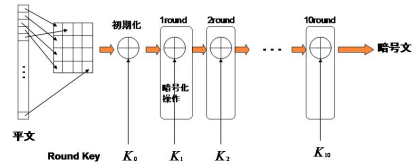
本研究ではブロック長 128bit, 鍵長 128bit を使用する.

2.2 暗号化アルゴリズム

AES(ブロック長 128bit) の暗号化アルゴリズムは以下の通り.

1. 入力された平文を 8bit 毎に区切り,4 × 4 マス (state) に振り分ける.
2. 入力された鍵を KeyExpansion 関数を用いて【規定ラウンド数+1】(11) 個に拡張.
3. state に対して AddRoundKey 変換を施す.
4. state に対し,SubByte 変換,ShiftRow 変換,MixColumn 変換,AddRoundKey 変換を【規定ラウンド数-1】(9) 回繰り返し行う.
5. 最終ラウンドのみ,SubByte 変換,ShiftRow 変換,AddRoundKey 変換だけを行う.

AESアルゴリズムの流れ(128bitの場合)



2.3 それぞれの変換について

AES は以下で示す 4 つの変換で成り立っている.

2.3.1 SubByte 変換 (SB)

8bit b_0,b_1,$\cdots$,b_7 をひとまとまりとして,0,1 係数の多項式 $b_0 + b_1x + \cdots + b_7x^7$ とみなし, 次式により体にする.
$GF(2^8) \simeq F_2[x]/(x^8 + x^4 + x^3 + x + 1)$
加法はビットごとに XOR をとり、乗法は多項式としてかけて同値関係で割る.(単位元は 1, 零元は 0)

SubByte 変換ではまず, $GF(2^8)$ での乗法の逆元をとる. ただし,0 は 0 に写す. この逆元処理を A とおく. 次に行列処理 B を行う.

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix}$$

最後にベクトル $(11000110)^T$ を F_2 上で足し算する. このベクトル処理を C とおく.

2.3.2 ShiftRow 変換 (SR)

$$\begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \Rightarrow \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{11} & a_{12} & a_{13} & a_{10} \\ a_{22} & a_{23} & a_{20} & a_{21} \\ a_{33} & a_{30} & a_{31} & a_{32} \end{pmatrix}$$

行ごとにかき混ぜ,i 行目 (i=0,1,2,3) を i バイト左にシフトする.

2.3.3 MixColumn 変換 (MC)

各列 $(a_0a_1a_2a_3)^T$ をバイト係数多項式 $a(x) = a_0 + a_1x + a_2x^2 + a_3x^3$ とみなし, それぞれに $GF(2^8)[x]/(x^4+1)$ 上で $c(x) = 3x^3 + x^2 + x + 2$ をかける.

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

2.3.4 AddRoundKey 変換 (ARK)

鍵を知らないとできない唯一の変換
共有する鍵情報:128 ビットの鍵を簡単な変換で 128 ビット 11 個に拡張し、それぞれのデータに加える. そのとき k 回目の変換は k 番目の 128 ビットを加える.

3 評価方法

3.1 χ^2検定

$$x^2 = \sum_{i=1}^{m} \frac{(O_i - E_i)^2}{E_i}$$

O_i:観測度数
E_i:期待度数
m:グループ数

期待度数と観測度数の差が大きいと χ^2 値も大きくなる. 上記によって得られた χ^2 値を表に示す.
有意水準は 0.05 とする.

4 SubByte 変換の乱数性の評価

4.1 検定方法

8 ビット中の上位 3 ビットを 0 に固定した 2^5 個のバイトの SubByte 変換後の 8 ビットに現れる 1 の個数について χ^2 検定を行う.
ラウンド数 1〜規定回数-1 回まで観測

4.2 結果

棄却される所を**太字**,r:ラウンド数とする.
逆元, 行列, ベクトルのそれぞれの処理を A,B,C とする.
このときの χ^2 値は, 自由度 3,$\chi^2 \geq 7.81$ で棄却

	使用した変換		
r	A	B と C	A と B と C
1	4.5961	5.2138	4.9675
2	**12.9613**	1.4401	4.2702
3	4.5961	0.1250	4.5776
4	**12.9613**	**12.9613**	3.9204
5	4.5961	5.2138	4.2046
6	**12.9613**	1.4401	0.5536
7	4.5961	0.1250	4.9667
8	**12.9613**	**12.9613**	5.8818
9	4.5961	5.2138	4.2416

表 1: 8 ビットにおける SB 変換の χ^2 値

逆元変換のみ, 行列とベクトル変換だけでは乱数性は低いが, 2 種類交互に繰り返し混ぜると乱数性が高くなることがわかった.

5 AES の乱数性の評価

5.1 検定方法

128 ビットを 8 ビットごとに分けた 16ヶ所の中に 1ヶ所だけに 00000001 を入れ, それ以外は全て 0 とし,AES 変換後の写った先の 1 の入り方について χ^2 検定を行う.
鍵 0(すべて 0 のビットのみ)
ラウンド数 1〜規定回数-1 回まで観測

5.2 結果

SubByte 変換を SB, ShiftRow 変換を SR, MixColumn 変換を MC, AddRoundKey 変換を ARK とおく.
このときの χ^2 値は, 自由度 15,$\chi^2 \geq 25.0$ で棄却
別の検定方法による先行研究でも、4 種類の変換のどのひとつを除いても乱数性が低いことが知られていたが, 本研究でも同じ結果が得られた. さらに本研究では,SubByte 変換の一部を除いたときの乱数性の評価を行った.

		除去した変換			
r	AES	SB	SR	MC	ARK
1	**3418**	**2436**	**3418**	**5186**	**1888**
2	**66.5**	**945**	**464**	**546**	**320**
3	17.0625	**279.5**	**170**	**171.5**	**208**
4	14.6875	**116**	**185.25**	**264.4375**	**528**
5	**35.9375**	**33.375**	**117.375**	**255.8125**	**176**
6	21.75	16.75	**140.5625**	**166.625**	**256**
7	15	**94**	**32.0625**	**181.5**	**288**
8	10	**235.875**	**99**	**104.0625**	**160**
9	**25.5625**	**210**	**185.25**	**377.5625**	**320**

表 2: AES のそれぞれの変換後の χ^2 値

	使用した変換				
r	なし	A	B と C	B	C
1	**2436**	**4868**	**3418**	**3418**	**4868**
2	**945**	**1457**	**27.5**	**116.5**	**1433**
3	**279.5**	**154.0625**	23.25	**36**	**896**
4	**116**	**27.9375**	15	16.25	**191.25**
5	**33.375**	12.9375	19.375	9.25	**66.75**
6	16.75	21.75	13.875	17	**42**
7	**94**	15.8125	**30.5**	10.25	**229.75**
8	**235.875**	6.875	18.25	3.75	**99.5**
9	**210**	19.6875	9.5625	**25.9375**	**165.125**

表 3: SB 変換の中身に着目した χ^2 値

6 まとめと今後の課題

先行研究と同様に 4 種類の変換のどのひとつを除いても乱数性が低いことがいえる. 今回着目した SubByte 変換の中の変換, 逆元, 行列, ベクトルについてみると, 行列を行う時、乱数性が高いとわかる. 一方, ベクトルのみを行うと何もしないよりも乱数性が低いとわかった.
AES に対して有効だった今回の評価方法を用いて, 他の暗号と AES の比較を行っていきたい.

参考文献

[1] 福田恵子,"共有鍵暗号方式の評価と比較", お茶の水女子大学修士論文,2010.

[2] J.Daemen,V.Rijmen," AES Proposal:Rijndael ", AES submission,1998.

[3] Oliver Pretzel,"Error-Correcting Codes and Finite Fields",1992.

[4] 篠崎信雄,"統計解析入門", サイエンス社,1994