通信を必要としない鍵更新方法を用いた共有鍵暗号の強化

理学専攻　情報科学コース　加藤知美

1　はじめに

共有鍵暗号を用いて安全に通信するためには，定期的に鍵交換を行う必要がある．しかしよく使われるディフィー・ヘルマンの鍵交換方法などの公開鍵暗号を用いた鍵交換には通信を必要とするので回数を増やしにくいことが多い．ここでは通信を必要としない代わりに共有秘密鍵情報を鍵のサイズより大きくとることを特徴とする鍵更新方法を提案し，標準暗号 AES の具体的な鍵更新方法の例について安全性の比較を行なった結果を紹介する．

2　AES(Advanced Encryption Standard) のしくみ

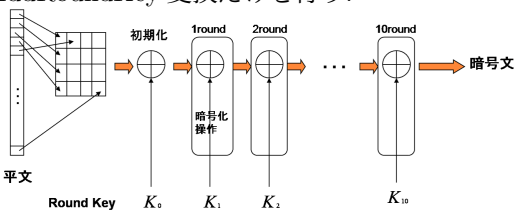
2.1　仕様

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．



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 (Sbox とも呼ばれる)

8bit b_7,b_6,$\cdots$,b_0 をひとまとまりとして,0,1 係数の多項式 $\cdots + b_7x^7 + b_1x + b_0$ とみなし, 次式により体にする．$GF(2^8) \simeq F_2[x]/(x^8 + x^4 + x^3 + x + 1)$
加法は係数ごとに GF(2) の元として計算し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 に移す. 次に以下のように処理を行う．

$$\begin{pmatrix} a_7 \\ a_6 \\ a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} b_7 \\ b_6 \\ b_5 \\ b_4 \\ b_3 \\ b_2 \\ b_1 \\ b_0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

2.3.2　ShiftRow 変換

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変換

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

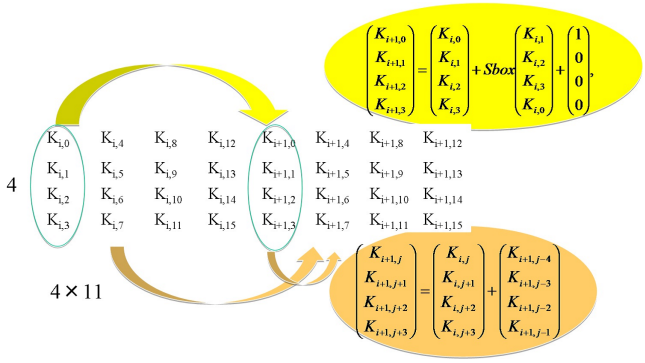
2.3.3　MixColumn 変換

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

鍵を知らないとできない唯一の変換
共有する鍵情報 128bit の鍵を簡単な方法で 128bit11 個に拡張し，それぞれの 128bit のデータに加える．そのとき k 回目の変換は k 番目の 128bit に加える.4 回に 1 回上の式を使い、それ以外は下の式を使う。



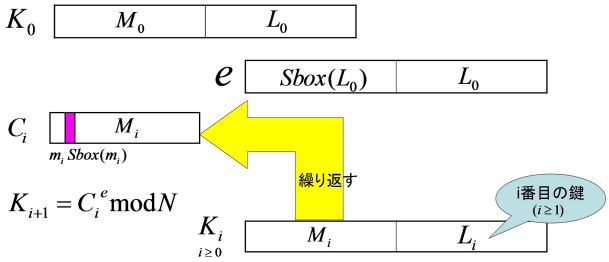
3　鍵更新方法

共有秘密情報を鍵のサイズより大きくとり，鍵更新ではそれを一方向性関数で更新し，更新した共有秘密情報の一部を新しい暗号鍵として用いることで，通信せず暗号鍵更新を行う．共有秘密情報の一部しか暗号鍵として用いない構成とすることで，安全性を損なうことなく鍵更新ができる．この方法では暗号鍵更新のための情報を通信しないため，その情報を漏らさないための処理が不要となり，暗号鍵更新のための計算量も通信を利用して新たな鍵に交換するよりも少なくで

きる．これにより頻繁に共有鍵を取り替えることが可能となり，同コストでこれまで以上に信頼性の高い暗号通信装置を実現できることが期待される．例としてAES の 128bit 鍵を以下のように更新する鍵更新方法について評価を行なう．

3.1 例

ここでは共有する秘密情報を 256bit とし，AES の鍵としてそのうちの 128bit を用いる．



最初に共有する 256bit の秘密情報 K_0 の下位 128bit を取り出したものを L_0、L_0 を Sbox したものを上位 128bit にした計 256bit を e とする

$i \geq 0$ について，K_i の上位 128bit を M_i，下位 128bit を L_i とする．M_i の最上位 8bit の m_i を Sbox し，その次の 8bit に上書きしたものを C_i とし，次式で新しい K_i を作成する．K_{i+1}=C_i^emodN

$i \geq 1$ のとき K_i の下位 128bit の L_i を i 番目の鍵として使う．

4 法 N の選び方

法 N は N=s × t で構成される 256bit の合成数で，s,t,(s-1)/2,(t-1)/2 は素数となるように選ぶ．この例では，以下を用いた．

s=10100101001110111001110101101010011000100011001010111011011111000000000011111111000001001000011100001010011011001111001011110101111(128bit)

t=11100001001110000111111111100101110110011010010111111111111011000111000000011001010111011100010001100011001010100111011110001000111111(128bit)

5 AES の乱数性の評価

AES は 4 種類の変換の一部をやめたりラウンド数を少なくしたりすると，十分に混ざらなくなることが知られている．以下で，平文・鍵ともに 128bit の AES を簡略化して意図的に安全でなくした暗号と，そこに鍵更新方法を適用した暗号の混ざり具合を比較することで乱数性を評価した結果を紹介する．

5.1 検定方法

128bit の平文を 8bit × 16ヶ所とみなす．この 16ヶ所の中の 1ヶ所だけに 00000001 を入れ，それ以外は全て 0 とした 16 通りの平文を変換した後の各 4bit の数について χ^2 検定を行う．

簡略化した暗号化の変換が十分によく混ざるものであれば，変換後の 16 × (128/4) 個の 4bit の数の集合はランダムに選んだ 4bit の数の集合とみなせるはずである．

共有秘密情報の初期値はすべて 0 とする．

ラウンド数 1〜規定回数-1 回まで観測

5.2 χ^2検定

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．

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

O_i：観測度数
E_i：期待度数
m：グループ数

5.3 結果

変換を表では以下の番号で表すことにする
1.SubByte 変換 (Sbox でも表せる)
2.ShiftRow 変換
3.MixColumn 変換
4.AddRoundKey 変換 (通常の鍵)
4'.AddRoundKey 変換 (提案手法での鍵)
このときの χ^2 値は，自由度 15，有意水準 0.05,$\chi^2 \geq 25.0$ で棄却

	1	2	3	4	5	6	7	8	9
1+2+3+4	3418	66.5	17.06	14.68	35.93	21.75	15	10	25.56
2+3+4	2436	945	279.5	116	33.37	16.75	94	235.8	210
1+3+4	3418	464	170	185.2	117.3	140.5	32.06	99	185.2
1+2+4	5186	546	171.5	264.4	255.8	166.6	181.5	104.0	377.5
1+2+3	1888	320	208	528	176	256	288	160	320
1+2+4'	10.12	8.562	8	14.5	10.81	17.81	19.87	15.18	13.93
1+3+4'	18.43	14.37	13.75	21.56	18.81	16.93	12.06	12.43	10.56
2+3+4'	25.75	9.062	27.37	26.12	9.437	17.06	8.312	18.81	14.06
1+4'	21	15	16.43	11.87	17.18	20.06	9.187	15	7.562
2+4'	31.5	20.5	9.437	7.437	11.12	17.31	9	19.5	18.5
3+4'	11.5	16.62	22.06	15.81	21.93	8.562	13.93	6.75	20
4'	11.81	9.937	20.93	14.18	10.06	9.875	16.62	16.75	21.43

縦軸：行った処理
横軸：ラウンド数
有効数字は4桁(切捨て)で表示 色つき の部分は棄却された

6 まとめと今後の課題

AES の 4 つの変換から 1 つ変換を抜くと乱数性が低下することが表よりわかるが，今回提案した鍵更新を用いれば，ずっと同じ鍵を使い続けるより乱数性が高くなることが示された．また，暗号に用いた 128bit 鍵 L_i についての情報が第三者に伝わった場合にも，256bit の秘密情報 K_i を予想するには残りの 128bit を予想する必要があり，これは次の 128bit の AES 鍵 L_{i+1} を予想するのと同じ難しさがあることがわかる．このため，この共有秘密情報を鍵サイズより大きくとる鍵更新方法を用いれば，通信を用いることなく，新規の鍵を選びなおすことと同等の安全性を確保できると期待される．

今回有効だった検定方法だけでなく，違う検定方法を用いて評価をしたい．

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

[5] 学校法人慶應義塾，萩田真理子，暗号鍵更新方法，特許第 3695526 号，特開 2003-110540，2003.