# 共有鍵暗号方式の評価と比較

福田恵子　(指導教員：萩田真理子)

## 1　研究背景

近年，通信において安全にデータの送受信を行うために様々な暗号アルゴリズムが用いられており，特に共有鍵暗号方式は大量のデータを送受信する際に利用されている．共有鍵暗号方式には多くの種類があるが，その暗号アルゴリズムの安全性が重要視されている．本研究では，暗号の安全性の評価手法の一つとして，暗号化関数をモデル化し統計的検定によって乱数性の高さを評価することを提案する．評価対象として暗号アルゴリズム Rabbit を用い，この結果と標準規格である AES(Advanced Encryption Standard) の結果を基に評価を行い，この手法の有効性の検証を行う．

## 2　Rabbit

Rabbit とは，ストリーム暗号の 1 つである．ECRYPT による eSTREAM Project に選ばれた暗号方式のうちの 1 つとなっている．Internal state は 513bit からなり，$8 * 32$ bit の state variables $x_{j,i}$ ，$8 * 32$ bit の counter variables $c_{j,i}$ ，1bit の counter carry $\phi_{7,i}$ で構成されている．128bit の秘密鍵 $K$ と 64bit の $IV$ から擬似乱数生成アルゴリズムを用いて擬似乱数を生成し，128bit ずつに区切った平文と排他的論理和をとることにより，暗号化または復号を行う．Rabbit の擬似乱数生成アルゴリズムは以下の通り．

1. 秘密鍵 $K$ を用いて $x_{j,i}$, $c_{j,i}$ を初期化（Key Setup Scheme）
2. $c_{j,i}$ を $IV$ を用いてさらに変換（IV Setup Scheme）
3. Counter System で次の $c_{j,i}$ を生成
4. Next-state Function で次の $x_{j,i}$ を生成
5. $x_{j,i}$ を用いて暗号化復号用擬似乱数 $s_i$ を生成（Extraction Scheme）
6. $s_i$ と暗号化または復号したいテキスト 128bit を XOR
7. 3〜6 を繰り返し，暗号化または復号を逐次行う

## 3　AES(Advanced Encryption Standard)

AES とは，2001 年 3 月，J.Daemen と V.Rijmen が提案した Rijndael が元となった暗号アルゴリズムである．アルゴリズムは，SubByte 変換，ShiftRow 変換，MixColumn 変換，AddRoundKey 変換の 4 つの関数に分けられており，SubByte 変換は byte 変換，SubByte 変換は行変換，MixColumn 変換は列変換，AddRoundKey 変換は RoundKey との排他的論理和をとる変換となっている．本研究では Rabbit と比較するため，ブロック長・鍵長ともに 128bit のものを使用しており，このときラウンド数は 10 回となっている．

## 4　AES に対する評価方法と結果

安全性の高い暗号の条件の一つとして，類似性の高い平文が類似性の低い暗号文へと変換されることが挙げられる．これは，類似した平文から類似した暗号文が生成されてしまうと，解読したい暗号文に類似した暗号文を用いて平文を推測されてしまう恐れがあるからである．したがって，暗号文と真性乱数が識別不可能であることが必要とされる．

本研究は Rabbit と比較するために，まず上記の条件を AES が満たしているかについて，いくつかの評価を行った．評価方法と条件については以下の通りである．

- 各暗号文を 4byte または 2byte 毎のブロックに区切り，暗号文 3 つずつに対して，一つのブロックを $(x, y, z)$-座標として三次元空間にプロット．
- 各暗号文の 1 のビットの数を数え，その分布と同数の乱数の 1 のビットの数の分布を比較．
- 各暗号文を 4byte 毎に区切って格納し，格納時に以前に等しい値を格納していた場合を衝突 1 としてその回数を計測．

どの評価方法においても，平文は 1 のビットが 1 箇所，もしくは 2 箇所のものを使用し，鍵はすべて 0 のものを用いて検証した．また，3 つ目の評価方法は，空間全体に $2^{32} = 4294967296$ 個の元が存在することより，衝突回数の期待値は多くとも $\sum_{k=1}^{33024} \frac{k-1}{2^{32}} = 0.126957$ となり，高々1 回程度である．

どの評価の結果も，1 ラウンドや 2 ラウンド目までは同数の乱数には見られない偏りが見られたが，3 ラウンド目以降は偏りが見られず，同数の乱数で行った結果と差が見られなかった．3 つ目の評価方法においても，1,2 ラウンド目は多数の衝突が確認されたが，3 ラウンド目以降は衝突が確認されなかった．よって，今回の評価方法においては，AES は乱数性が高いという結果が得られた．

## 5　AES の各関数に対する評価

ここで，AES が，SubByte 変換，ShiftRow 変換，MixColumn 変換，AddRoundKey 変換の 4 つの関数から成り立つことに注目し，AES の 4 つの変換のうち，どの変換が重要な変換であるかを検証した．AES の SubByte 変換 (SB)，ShiftRow 変換 (SR)，MixColumn 変換 (MC)，AddRoundKey 変換 (ARK) の 4 つの変換のうち，どれか一つの変換を除いて暗号化を行い，衝突数を計測した．

表 1 を見てわかるように，どの変換を抜いた場合においても衝突回数が格段に増加している．したがって，AES の 4 つの変換は，どの変換を抜いた場合でも乱数性が非常に悪くなることがわかった．また，ShiftRow 変換と MixColumn 変換それぞれを除いた場合は等しい結果となり，これは各行 (列) に出現する種類の総

表 1: 除去した変換毎の衝突数

| Round | AES | 除去した変換 | | | |
|---|---|---|---|---|---|
| | | SB | SR | MC | ARK |
| 1 | 32495 | 32495 | 32495 | 32495 | 32495 |
| 2 | 25266 | 28706 | 31966 | 31966 | 29145 |
| 3 | 0 | 0 | 30908 | 30908 | 24770 |
| 4 | 0 | 14396 | 30908 | 30908 | 24768 |
| 5 | 0 | 14396 | 30908 | 30908 | 24768 |
| 6 | 0 | 0 | 30908 | 30908 | 24768 |
| 7 | 0 | 4496 | 30908 | 30908 | 24768 |
| 8 | 0 | 30908 | 30908 | 30908 | 24768 |
| 9 | 0 | 30908 | 30908 | 30908 | 24768 |
| 10 | 0 | 24388 | 30908 | 30908 | 24768 |

数分を全体 (33024) から除いたものである．具体的には，3 ラウンド目以降は，同一の行 (列) に 1 のビットが 2 か所立っているものは 1 種類しかなく，これは ${}_{32}C_2 * 4 = 1984$ 種類，また各行 (列) に 1 のビットが 1 か所立っているものの種類は ${}_{32}C_1 * 4 = 128$ 種類，全て 0 の場合は 4 種類あるので，計 2116 種類が出現する．これは全体 33024 から衝突回数 30908 を引いたものと等しい．この原因は，同一のものは変換後も同一のものになるためである．

以上の結果から，AES の乱数性は高く，10 回のラウンド数は安全性を高めるために十分であることがわかった．そしてその安全性は 4 つの変換を全て組み合わせることに起因しており，全ての変換が不可欠であるという結果が得られた．

## 6 Rabbit に対する評価方法と結果

次に，本研究はストリーム暗号の Rabbit についても同様の乱数性の高さに対する評価を行った．AES 同様，3 次元空間にプロットする方法，1 のビットの数の分布，衝突数などを同数の乱数の分布と比較した．どの評価方法においても，秘密鍵は 1 のビットが 1 箇所，もしくは 2 箇所のものを使用し，IV は全て 0 として使用した．

どの評価方法においても，第 1 キーストリームには偏りが見られたものの，第 2 キーストリーム以降は偏りが見られなかった．これにより，今回の 3 つの評価方法においては，AES よりも Rabbit の方が乱数性が高いということがわかった．これは，Rabbit がストリーム系の暗号方式であり，高速に暗号化を行う性質をもつことに起因する結果であると考えられる．

## 7 $\chi^2$ 検定を用いた評価方法と結果

さらに，本研究では $\chi^2$ 検定を用いて AES と Rabbit の乱数性を評価した．評価条件は前述と同様の条件を用い，上位 5 ビットの衝突を計測し，真性乱数に対して $\chi^2$ 検定を行った．

どちらもラウンド数が少ないと $\chi^2$ 検定において棄却されることがわかるが，ラウンドを進めるにつれて偏りがなくなっていることがわかった．$\chi^2$ 検定を用いた評価方法においても，AES，Rabbit ともに乱数性が高いという結果が得られた．

本研究では，さらに $\chi^2$ 検定を用いて AES の関数を 1 つ除去した場合や，Rabbit の Counter System 内の定数 a を全て 0 とした場合においても評価を行った．

表 2: AES と Rabbit の $\chi^2$ 値の確率

| Round | $AES$ の$\chi^2$ 値 | $Rabbit$ の$\chi^2$ 値 |
|---|---|---|
| 1 | 0.000000 | 0.000000 |
| 2 | 0.023914 | 0.977118 |
| 3 | 0.924614 | 0.999986 |
| 4 | 0.995691 | 0.98398 |
| 5 | 0.919061 | 0.983831 |
| 6 | 0.997007 | 0.99976 |
| 7 | 0.999996 | 0.998325 |
| 8 | 0.974524 | 0.979105 |
| 9 | 0.999726 | 0.994357 |
| 10 | 0.998867 | 0.997959 |

表 3: AES と Rabbit の関数や定数を除去した場合の $\chi^2$ 値の確率

| Round | $AES(SB$ 除去) | $Rabbit$(定数 $a$ が 0) |
|---|---|---|
| 1 | 0.000000 | 0.000000 |
| 2 | 0.000000 | 0.000000 |
| 3 | 0.000000 | 0.000000 |
| 4 | 0.000000 | 0.000000 |
| 5 | 0.000000 | 0.000000 |
| 6 | 0.00038 | 0.000000 |
| 7 | 0.000000 | 0.000000 |
| 8 | 0.000000 | 0.000000 |
| 9 | 0.000000 | 0.000000 |
| 10 | 0.000000 | 0.000000 |

どちらも $\chi^2$ 値の確率が 0 に近く，関数や定数を 1 つ除去しただけで乱数性が低くなるという結果が得られた．

## 8 まとめ

結果，本研究においては AES より Rabbit の方が乱数性が高いということがわかった．また，AES，Rabbit ともに効率の良い暗号アルゴリズムが用いられており，関数を 1 つ除去した場合でも乱数性が著しく低くなることが観測された．

## 参考文献


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

[2] NIST, "Advanced Encryption Standard (AES) ", FIPS PUB 197, 2001.

[3] Martin Boesgaard, Mette Vesterager, Thomas Christensen and Erik Zenner, "The Stream Cipher Rabbit", eSTREAM, ECRYPT Stream Cipher Project, Report 2005/024, http://www.ecrypt.eu.org/stream, (2005).