悉無律をみたす暗号利用モードの設計と評価

赤尾　奏名汰[†1,a)]

概要：ブロック暗号により暗号化で用いる鍵の全数探索を困難にさせるための方法として、様々な研究がなされている。先行研究によると全ての分散片を集めない限り復元できず 1 つでも足りない場合は何の情報も得られない、つまり全か無かの法則を持つアルゴリズムを、悉無律を満たすアルゴリズムという。これらの暗号アルゴリズムは暗号利用モードのカウンタモードを基に構成されているが、他のモードについては言及されていない。本論文では、Rivest が提案した パケット変換に基づき、悉無律を満たす暗号利用モードの設計を行った。また、C#上でプログラムを動作させ、処理時間を比較し評価を行った。

キーワード：共通鍵暗号、ブロック暗号、暗号利用モード

Design and Evaluation of Block Cipher Modes of Operation to Achieve All-Or -Nothing Rule

SONATA AKAO[†1,a)]

Abstract: Various researches have been carried out as a method for making the exhaustive search of keys used in encryption difficult by block encryption. Previous research has shown that an algorithm that satisfies all or none of the rules is called an algorithm that satisfies All-Or-Nothing Rules. These cryptographic algorithms are based on the counter mode of the cipher usage mode, but other modes are not mentioned. In this paper, was carried out design of an eblock cipher modes of operation satisfying all-or-none based on packet conversion proposed by Rivest. The program was run on C#, and the processing time was compared and evaluated.

Keywords: symmetric cipher, block cipher, block cipher modes of operation

1. 研究背景

1.1 共通鍵暗号と公開鍵暗号

我々は，インターネットなどを通してさまざまな場面で情報を送受信している．その際，誰でもアクセスが出来る通信路を用いているため，第三者にデータを盗聴されたり，改竄されたりすることが出来ないようにデータを暗号化して送信する必要がある．受信者は暗号化されたデータを元のデータに戻す復号という処理を行い，第三者に対して秘密にしたままデータの交換を行う．暗号化や復号では，暗号アルゴリズムと鍵が用いられる。現代では暗号アルゴリズムは公開され鍵は秘密にされることが一般的である．暗号アルゴリズムを公開することで，安全性の強度を学界などで議論できるだけでなく，標準化が可能になる．

このような暗号化方式は共通鍵暗号と公開鍵暗号に分けられる．共通鍵暗号は暗号化と復号で同じ鍵を使う方式であり，通信の前に送信者と受信者で秘密に同一の鍵を認識しておく必要がある．このときに使用する鍵の事を共通鍵という．一方，公開鍵暗号は暗号化に用いる鍵と復号に用いる鍵が異なる方式である．暗号化に用いる鍵を公開鍵，復号に用いる鍵を秘密鍵という．受信者が公開鍵と秘密鍵のペアを生成し，公開鍵のみを公開する．送信者は公開鍵を用いてデータを暗号化し送信する．受信者は秘密鍵を用いて暗号文を復号する．ここで暗号化されたデータは秘密鍵によってのみ復号でき，公開鍵から秘密鍵を求めることは困難である必要がある．

[†1] 現在，九州大学
Presently with Kyushu University
[a)] akao.sonata.598@s.kyushu-u.ac.jp

1.2 ストリーム暗号とブロック暗号

暗号アルゴリズムはストリーム暗号とブロック暗号に大別される．ストリーム暗号とは平文を 1 ビットや 1 バイトなど短い単位で暗号化するものである．平文のビット列 M=(b_1,b_2,$\cdots$) を鍵のビット列 K=(k_1,k_2,$\cdots$) を用いて，c_1=$b_1 \oplus k_1$,c_2=$b_2 \oplus k_2$,$\cdots$と暗号化する．ここで C=(c_1,c_2,$\cdots$) が暗号文となる．鍵系列 K が真にランダムな場合は、無限大の能力を有する敵に対しても安全である．しかし，鍵系列 K を真にランダムにするには，平文ごとに K を新しく選び直さなければならないため効率が悪い．そこで，線形シフトレジスタなどを利用し，短い鍵から長い擬似ランダムな鍵系列を生成する方法が提案されている．

ブロック暗号とは，平文のビット列をブロック長と呼ばれるある一定の長さ n ごとに分割し，そのブロック単位で暗号化を行うものである．米国においては，公募の上，1977 年に DES 暗号が連邦政府の標準暗号として制定された．しかし，コンピュータの飛躍的向上により，DES 暗号が安全でなくなる懸念のため，公募の上，2001 年に AES 暗号が新しい標準暗号として制定された．

1.3 ブロック暗号利用モード

ブロック長 n よりも長い平文 M を暗号化する際には，ブロック暗号 E を用いて暗号化できるように，暗号利用モードが使用される．各種利用モードを以下に示す．

ここで，平文ブロック M_i を鍵 K を用いて暗号化アルゴリズム Enc で暗号化し，暗号文ブロック C_i を得るとする．また，Dec は復号アルゴリズムである．つまり次式が成り立つ．

$$M_i = Dec_K(C_i) = Dec_K(Enc_K(M_i))$$

イ．ECB モード

ECB モードは，最も簡単な利用モードである．平文を n ビットごとのブロックに分割し，各ブロックを独立にブロック暗号の暗号化関数の入力とする．その結果，得られた出力が暗号文ブロックとなり，それらを繋げたものが暗号文となる．

暗号化

$$C_i = Enc_K(M_i)$$

復号化

$$M_i = Dec_K(C_i)$$

ECB モードでは独立にブロックを暗号化するため，同じ鍵を用いた場合，パターンが検出されてしまう．つまり、$M_i = M_j$ であるならば，$C_i = C_j$ となる．

ロ．CBC モード

CBC モードでは，平文を n ビットごとのブロックに分割し，各ブロックは前の暗号文との XOR を取ってから暗号化される．すなわち，各々の暗号文ブロックはそれ以前のすべての平文ブロックに依存すること

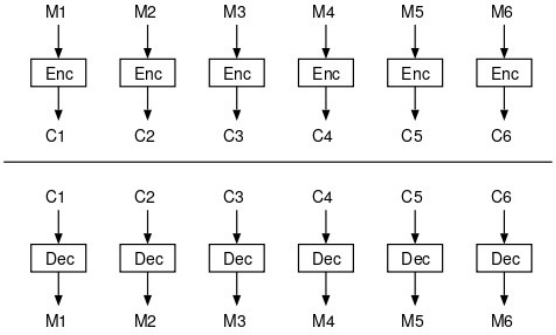

図 1 ECB モードの暗号化と復号化
Fig. 1 Encryption and decryption of ECB mode.

となる．最初のブロックの暗号化には初期化ベクトル ($C_0 = IV$) が用いられる．

暗号化

$$C_i = Enc_K(M_i \oplus C_{i-1})$$

復号化

$$M_i = Dec_K(C_i) \oplus C_{i-1}$$

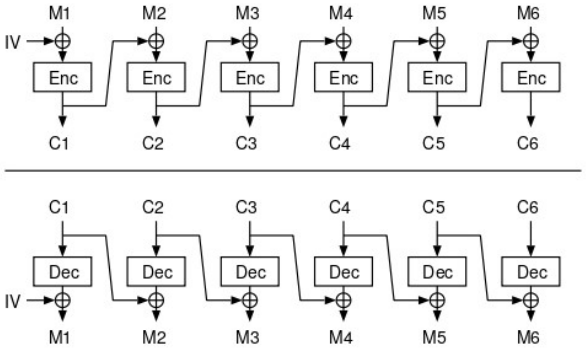

図 2 CBC モードの暗号化と復号化
Fig. 2 Encryption and decryption of CBC mode.

CBC モードは，ECB モードの欠点を補うもので，最も広く用いられている暗号利用モードである．ただし，各ブロックの暗号化にその前のブロックの暗号化の結果を使用することから暗号化処理を並列化することができないことは欠点である．復号化では，ブロック暗号処理に関する並列処理性は可能である．

ハ．CFB モード

CFB モードは，CBC モードと類似しており，ブロック暗号を自己同期型のストリーム暗号として扱うものである．CFB モードの操作は CBC モードとよく似ており，特に復号処理は CBC モードでの復号処理をほぼそのまま逆転させたものとなる．

暗号化

$$C_i = M_i \oplus Enc_K(C_{i-1})$$

復号化

$$M_i = C_i \oplus Enc_K(C_{i-1})$$

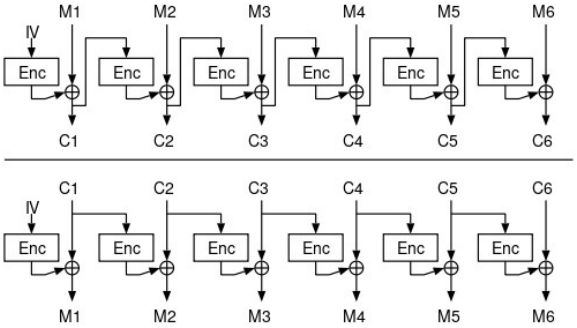

図 3 CFB モードの暗号化と復号化
Fig. 3 Encryption and decryption of CFB mode.

CFB モードでは，ブロック暗号の復号化関数を利用しない．よって，CFB 暗号化，CFB 復号化の両方の機能を実装する場合には，その実装コストは，ECB モードや CBC モードより軽いことが期待できる．

二．OFB モード

OFB モードは，任意のビット長の平文に対して処理ができるモードで，初期値のみに依存し逐次的に擬似乱数を生成して暗号化を行う．まず、平文を n ビットごとのブロックに分割し，最後の端数部分は端数ブロックとして扱う。内部レジスタの初期値を R_0 としてこの利用モードの初期値とする．R_{i-1} をブロック暗号の入力とし，その出力を R_i として平文ブロック M_i と排他的論理和することによって暗号文ブロック C_i を得る．

暗号化

$$R_i = Enc_K(R_{i-1})$$
$$C_i = M_i \oplus R_i$$

復号化

$$R_i = Enc_K(R_{i-1})$$
$$M_i = C_i \oplus R_i$$

OFB モードには，暗号化にも復号化にも比率処理性はない．また，CFB モード同様，復号化関数を利用しない．

ホ．CTR モード

CTR モードは，OFB モードとほぼ同様に暗号化を行うが，異なる点は，内部レジスタの初期値を R_0 とすると，次のブロックに進むたびに内部レジスタは整数カウンタとして 1 を数え上げる．つまり，$R_{i+1} = R_i + 1$ である．また、同一の鍵の下で安全に暗号化を行うために，初期値に対しては特別な運用が必要になる．

暗号化

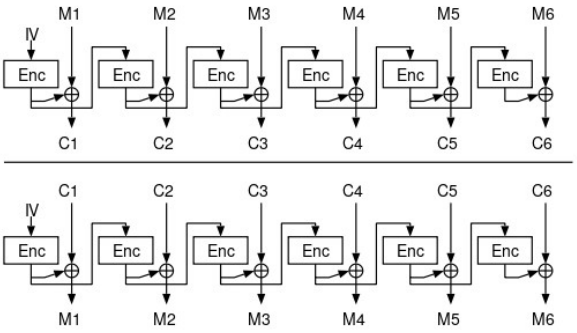

図 4 OFB モードの暗号化と復号化
Fig. 4 Encryption and decryption od OFB mode.

$$C_i = M_i \oplus Enc_K(R_i)$$
$$R_{i+1} = R_i + 1$$

復号化

$$M_i = C_i \oplus Enc_K(R_i)$$
$$R_{i+1} = R_i + 1$$

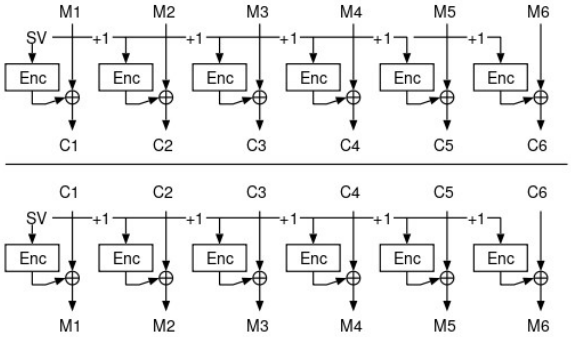

図 5 CTR モードの暗号化と復号化
Fig. 5 Encryption and decryption of CTR mode.

ヘ．2DEM モード (2D Encryption Mode)

2DEM モードは，ECB モードの安全性の懸念，CBC モードの並列処理の低さを克服することを目的に主にバイトデータを二次元配列で解釈し，暗号化処理を行うことを記述したものである．具体的には，まず，平文を ECB モードで処理したものをバイト単位でインターリーブする．そうしてできたブロック列を再度 ECB モードで処理し，その結果をインターリーブして暗号文でブロック列とするものである．

ト．ABC(Accumulated Block Chaining) モード

ABC（累積ブロック連鎖）モードは，エラー伝播が最後まで続くような暗号利用モードとして AES 利用モードに提案された．しかし，提案は秘匿の目的のみであり，上記性質が暗号学的な意味のある安全性には特に関連していない．ABC モードは 2 つの初期値を用いて暗号化を行う．中間値を H_i とする．

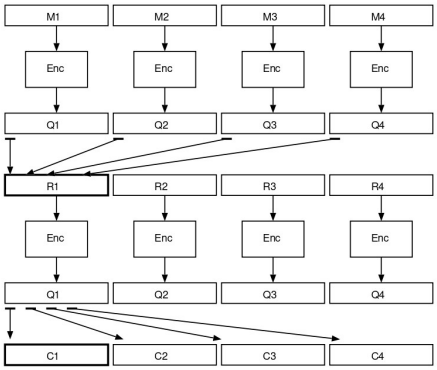

図 6 2DEM モードの暗号化と復号化
Fig. 6 Encryption and decryption of 2DEM mode.

暗号化

$$H_0 = IV_1, C_0 = IV_2$$
$$H_i = M_i \oplus H_{i-1} \oplus C_{i-1}$$
$$C_i = Enc_K(H_i) \oplus H_{i-1}$$

復号化

$$H_0 = IV_1, C_0 = IV_2$$
$$H_i = Dec_K(C_i \oplus H_{i-1})$$
$$M_i = H_i \oplus H_{i-1} \oplus C_{i-1}$$

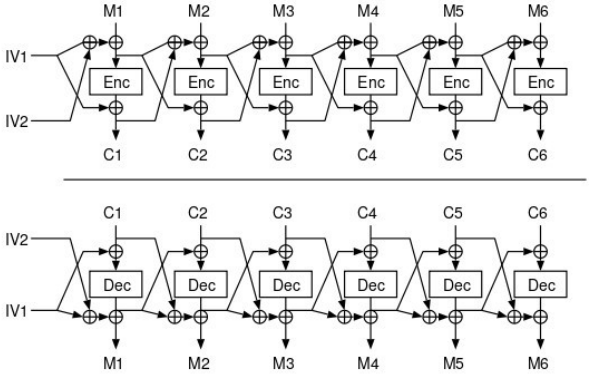

図 7 ABC モードの暗号化と復号化
Fig. 7 Encryption and decryption of ABC mode.

チ．IGE(Infinite Garble Extension) モード

IGE(無限改竄拡張) モードは，もともと CBC モードと同じくらい古くに提案された利用モードである．AES の利用モードで，このモードに対する解析結果が発表されメッセージ認証に対して安全でないことが示されている．暗号化と復号化の処理フローが同じ（上下対称）であるのは，何らかの実装の利点があるかもしれないが，それ以上に復号化処理でブロック暗号の復号関数が必要となり，そうでない CFB モード，OFB モード，CTR モードなどがより効率的である可能性が高い．

暗号化

$$M_0 = IV, C_0 = Enc_{K'}(M_0)$$
$$C_i = Enc_K(M_i \oplus C_{i-1}) \oplus M_{i-1}$$

復号化

$$M_0 = Dec_{K'}(C_0)$$
$$M_i = Dec_K(C_i \oplus M_{i-1}) \oplus C_{i-1}$$

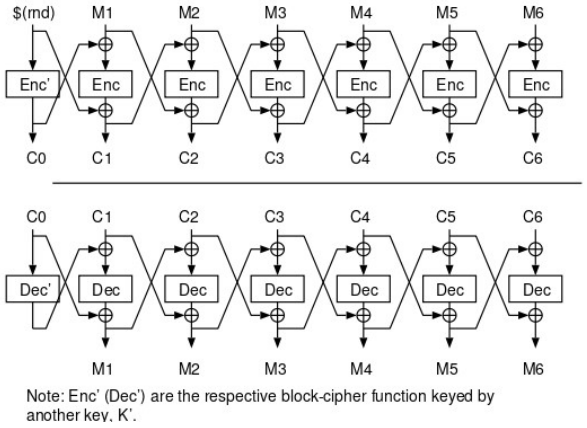

図 8 IGE モードの暗号化と復号化
Fig. 8 Encryption and decryption of IGE mode.

リ．F8@ 3GPP

3GPP では，ブロック暗号 KASUMI の利用モードとして F8(秘匿) と F9(認証) を定義している．F8 モードでは，仕様で定義された nonce 入力と鍵，カウンタ値から鍵ストリームが生成され，ストリーム暗号的に暗号化される．CTR モードと CBC モードを組み合わせたようなものである．安全性には問題はないようにみられるが，並列処理性能がない．

暗号化

$$H_0 = 0$$
$$H_i = Enc_K(Enc_K(nonce) \oplus i - 1 \oplus H_{i-1})$$
$$C_i = M_i \oplus H_i$$

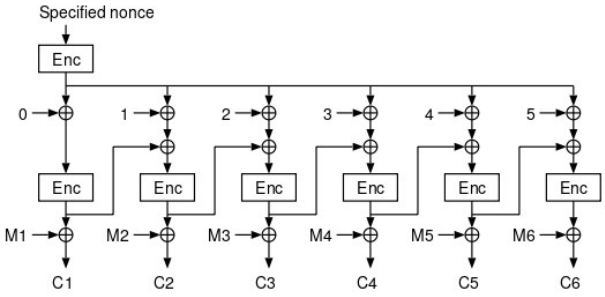

図 9 F8 モードの暗号化
Fig. 9 Encryption and decryption of F8 mode.

1.4 秘密分散共有法

秘密鍵の紛失を防ぐために，秘密鍵の複製を多数作成しそれぞれを別の場所で保存すると，盗難の危険性が増大してしまう．一方で，複製を少なくすると，その部分を集中的に攻撃されるリスクが高まったり，すべてを紛失してしまったりする危険性が増大する．これらの相矛盾する問題を改善する方法が秘密分散共有法である．例えば、Shmir の (k,n) 閾値秘密分散では，秘密 s から n 個の値を生成する．その n 個のうち任意の k 個の値が集まれば秘密 s を求めることが出来るが，k-1 個以下の値では秘密 s に関する情報を全く得ることが出来ない．

2. 先行研究

Rivest はブロック暗号利用モードで暗号化する前に平文データに対して前処理をする，新しい利用モードを提案した．その前処理の事を AONT(All-Or-Nothing Transform) といい、AONT を含めた新しい利用モードの事を All-or-nothing encryption mode という．このモードを使用することにより，平文の 1 つのブロックを求めるには暗号文全体を復号しなければならないため，攻撃者による鍵の全数探索が困難になる．

All-or-nothing encryption mode による暗号化の流れは図 2.1 に示す．まずは平文 M を AONT によって擬似平文 M' に変換する．次に，擬似平文 M' を入力として暗号利用モードにより暗号化することで暗号文 C を得ることが出来る．ここで，利用モードによる暗号化では CBC モードや CTR モードなどが用いられる．CBC モードを用いたときは，特に all-or-nothing CBC mode と言われる．

ここで $|M'| \geqq |M|$ を達成する．AONT の変換 $f : M \to M'$ は以下の事を満たす必要がある．

・変換 f は可逆である．
・f も f^{-1} も多項式時間で計算可能である．
・擬似平文 M' のいずれかのブロックが未知である場合，任意の平文 M のブロックも計算上求めることが出来ない．

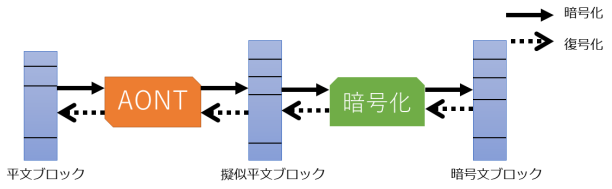

図 10 All-or-nothing encryption mode による暗号化の流れ
Fig. 10 Encryption flow by All-or-nothing encryption mode

Rivest は AONT を達成するアルゴリズムとして package transform を提案した．これの擬似コードをコード 1 に示す．ただし，平文 $M = (m_1, m_2, \cdots, m_s)$ とする．package transform では擬似平文のブロック数は平文のブロック数より 1 つ多くなる．つまり，Shamir の (k,n) 閾値秘密分散でいうと，(s+1,s+1) 閾値法ということが出来るだろう．

package transform はブロック暗号利用モードの CTR モードを元に構成されている．s 個の平文ブロックを CTR モードによって s 個の擬似平文ブロックを生成する．次に，s+1 個目の擬似平文ブロックは，CTR モードで用いた鍵と s 個の擬似平文ブロックの各ハッシュを排他的論理和をして生成される．このようにすることによって，鍵を求めるためには s+1 個すべての擬似平文ブロックを求めなければならない．

–Let the input message be $m_1, m_2, \cdots, m_s$.
–Choose at random a key K'
for the package transform block cipher.
–Compute the output sequence $m'_1, m'_2, \cdots, m'_{s'}$, for $s' = s + 1$ as follows:
Let $m'_i = m_i \oplus Enc_{K'}(i)$ for $i = 1, 2, 3, \cdots, s$.
Let

$$m'_{s'} = K' \oplus h_1 \oplus h_2 \oplus \cdots \oplus h_s,$$

where

$$h_i = Enc_{K_0}(m'_i \oplus i) \qquad \text{for } i = 1, 2, \cdots, s,$$

where K_0 is a fixed, publically-known encryption key.

コード 1:package transform

3. 悉無律をみたす暗号利用モード

3.1 設計方法

3.1.1 設計方法の提案

Rivest の提案した package transform は暗号利用モードの CTR モードを基にして構成されている．各ブロックを CTR モードによって擬似平文ブロックに変換した後，全ての擬似平文ブロックのハッシュと鍵を排他的論理和することによって，AONT を達成している．

この技法を使えば，どの暗号利用モードでも AONT，つまり悉無律を達成する変換を設計ことが出来る．つまり，平文ブロックを暗号利用モードによって擬似平文ブロックに暗号化し，その全てのブロックのハッシュと鍵を排他的論理和すればよい．

3.2 各モードのアルゴリズム

1.1.3 で示した暗号利用モードについて，平文ブロック m_i を擬似平文ブロック m'_i へ変換するアルゴリズムを示す．2.1 と同様にブロック暗号の暗号化アルゴリズムを Enc，K' をランダムに選んだ鍵，K_0 を公知の鍵とする．

イ．ECB モード

$$\text{for } i = 1, \cdots, n$$
$$m'_i = Enc_{K'}(m_i)$$

$$h_i = Enc_{K_0}(m'_i \oplus i)$$
$$m'_{n+1} = K' \oplus h_1 \oplus \cdots \oplus h_n$$

ロ．CBC モード

$$m'_0 = IV$$
for $i = 1, \cdots, n$
$$m'_i = Enc_{K'}(m_i \oplus m'_{i-1})$$
for $i = 0, \cdots, n$
$$h_i = Enc_{K_0}(m'_i \oplus i)$$
$$m'_{n+1} = K' \oplus h_0 \oplus h_1 \oplus \cdots \oplus h_n$$

ハ．CFB モード

$$m'_0 = IV$$
for $i = 1, \cdots, n$
$$m'_i = m_i \oplus Enc_{K'}(m'_{i-1})$$
for $i = 0, \cdots, n$
$$h_i = Enc_{K_0}(m'_i \oplus i)$$
$$m'_{n+1} = K' \oplus h_0 \oplus h_1 \oplus \cdots \oplus h_n$$

二．OFB モード

Inisial Value R_0
for $i = 1, \cdots, n$
$$R_i = Enc_{K'}(R_{i-1})$$
$$m'_i = m_i \oplus R_i$$
$$h_i = Enc_{K_0}(m'_i \oplus i)$$
$$m'_{n+1} = K' \oplus h_1 \oplus \cdots \oplus h_n$$

ホ．CTR モード

Rivest の提案した package transform が，CTR モードを基に構成されたものである．

Inisial Value R_0
for $i = 1, \cdots, n$
$$m'_i = m_i \oplus Enc_{K'}(R_0 + i)$$
$$h_i = Enc_{K_0}(m'_i \oplus i)$$
$$m'_{n+1} = K' \oplus h_1 \oplus \cdots \oplus h_n$$

ヘ．2DEM モード

簡単のため，2DEM モードの鍵 K の下での一連の暗号化処理を $\mathscr{E}_K$ と表すとする．また，便宜上，平文全体を M、擬似平文全体を M' とする（ブロック数は n）．

$$M' = \mathscr{E}_{K'}(M)$$
for $i = 1, \cdots, n$
$$h_i = Enc_{K_0}(m'_i \oplus i)$$
$$m'_{n+1} = K' \oplus h_1 \oplus \cdots \oplus h_n$$

ト．ABC モード

$$H_0 = IV_1, m'_0 = IV_2$$
for $i = 1, \cdots, n$
$$H_i = m_i \oplus H_{i-1} \oplus m'_{i-1}$$
$$m'_i = Enc_{K'}(H_i) \oplus H_{i-1}$$
$$m'_{n+1} = IV_1$$
for $i = 0, \cdots, n+1$
$$h_i = Enc_{K_0}(m'_i \oplus i)$$
$$m'_{n+2} = K' \oplus h_0 \oplus h_1 \oplus \cdots \oplus h_{n+1}$$

チ．IGE モード

このモードでは，公知の鍵 K_0 に加えて，鍵 K" もあらかじめ共有する必要がある．

$$m_0 = IV, m'_0 = Enc_{K"}(m_0)$$
for $i = 1, \cdots, n$
$$m'_i = Enc_{K'}(m_i \oplus m'_{i-1}) \oplus m_{i-1}$$
for $i = 0, \cdots, n$
$$h_i = Enc_{K_0}(m'_i \oplus i)$$
$$m'_{n+1} = K' \oplus h_0 \oplus h_1 \oplus \cdots \oplus h_n$$

リ．F8@ 3GPP

$$H_0 = 0$$
for $i = 1, \cdots, n$
$$H_i = Enc_{K'}(Enc_{K'}(nonce) \oplus i - 1 \oplus H_{i-1})$$
$$m'_i = m_i \oplus H_i$$
$$h_i = Enc_{K_0}(m'_i \oplus i)$$
$$m'_{n+1} = K' \oplus h_1 \oplus \cdots \oplus h_n$$

3.2.1 各モードの計算量の評価

上記で示した各モードのアルゴリズムについて，ブロック暗号を使用する数，排他的論理和をする数，平文ブロックに対する擬似平文ブロックの増加数を比較し，評価する．ブロック暗号を使用する数と排他的論理和をする数は計算量に関わり，ブロック数の増加数はメモリー容量に関わる．

モード	ECB	CBC	CFB	OFB	CTR
ブロック暗号	$2n$	$2n+1$	$2n+1$	$2n$	$2n$
排他的論理和	$2n$	$3n+2$	$3n+2$	$3n$	$3n$
ブロック数	$+1$	$+2$	$+2$	$+1$	$+1$

モード	2DEM	ABC	IGE	F8
ブロック暗号	$3n$	$2n+2$	$2n+2$	$2n+1$
排他的論理和	$2n$	$5n+4$	$3n+2$	$5n-1$
ブロック数	$+1$	$+3$	$+2$	$+1$

各モードを比較すると，ECB モードが 3 項目すべてにおいて効率の良い結果となっている．ECB モードは単純なアルゴリズムのため実装が簡単である一方，同一の入力に対して，同一の出力を得るという大きな欠点があり，安全性に問題がある．したがって，擬似平文ブロックの偏りから，鍵を推測しやすくなり，全ての擬似平文ブロックを集めなくても解読できる可能性が大きい．そのため ECB モードを基とする構成は，悉無律を満たすとはいえない．

計算量に関して，ブロック暗号と排他的論理和の使用数で比較したが，それぞれ 1 回あたりの実行時間はブロック暗号の方がかなり大きいと言える．従って，評価するにあたりブロック暗号の使用回数に注目する．ECB モード以外では，OFB モードと CTR モードが効率が良いと分かる．さらに，2 つのモードとも排他的論理和の数は等しく，メ

モリー容量に関するブロック数の増加数についても最小値となっている．

McEvoy らの考察では，CTR モードは平文ブロックとの計算の前にカウンタを計算することが出来る環境であるならば，事前に計算し値を保存しておくことで，実際に平文ブロックに対して処理する際の実行時間を短縮できるとされている．これは OFB モードも事前計算が可能であるアルゴリズムであるので当てはまる．

3.3 実行・比較方法

今回は，C#を用いて，ECB モード，CBC モードについてプログラムを実行した。github 上に公開されているソースコードを参考にコードを作成し，それぞれのモードについて実行時間を基に評価を行った．なお，AONT で変換する元データは，163,846kB の適当な zip ファイルを用いた．また，このコードでは，ブロックの数とその 1 つのブロックのサイズをパラメータとして指定できる．ただし，最後のブロックのサイズは調節されるため，指定したサイズより大きくなったり，小さくなったりする．今回の評価では，全てのブロックがほぼ同じ大きさになるように調節する．

また，基本的に元データのサイズはブロック数と各ブロックのサイズの積となるため，ブロック数と各ブロックのサイズは反比例の関係にある．そこで，各パラメータを以下の 4 パターンに設定して比較を行った．それぞれ 10 回ずつ実行時間を測定し，その平均値をとる．

・ブロック数 = 10、サイズ = 16384kB
・ブロック数 = 100、サイズ = 1638kB
・ブロック数 = 1000、サイズ = 163.9kB

3.4 結果

ECB モードの結果を図 3.1，CBC モードの結果を図 3.2 に示す．

3.5 考察

ECB モード，CBC モードのどちらも変換にかかる時間はブロック数が多くなるにつれ，長くなっている．これはブロック数が増えることで演算量も増加することが原因であると思われる．しかし，ブロック数が多いほうがブロック 1 つに対するリスクは小さくなるため，計算量と安全強度はトレードオフの関係にあると言える．

ECB モードと CBC モードの関係では，ECB モードのほうが単純なアルゴリズムであるため，実行時間も小さくなると予想されるが，ブロック数が 10 の場合は，CBC モードの実行時間のほうが小さかった．ただし，ブロック数の増加に対する実行時間の増加の割合は ECB モードのほうが小さい．

今回のプログラムの仕様上，package transform の基になっている CTR モードの実装が出来ず，それとは比べることは出来なかった．ただし暗号利用モードそのものの違いから，いくらか推測できる点がある．まず，逆変換 (復号) する際に ECB モードや CBC モードでは復号化関数が必要になる一方で，CTR モードでは復号化関数を利用しないため，実装コストは CTR モードの方が軽いと思われる．また，ECB モードはその他の 2 つのモードに比べ，簡単な構造であるが，その分の安全性は懸念される．暗号化として用いる暗号利用モードとしてしばしば使われるのは CBC モードであるので，前処理としても CBC モードを使用することは，実装が簡単になる．つまり，前処理と暗号化で用いる暗号利用モードは同じであれば，実装コストを比較的小さく抑えられると考えられる．

4. おわりに

現代社会では，年々コンピュータの性能が増大し，安全であるとされている暗号技術も次第に脆弱になることも考えられる．単純な攻撃方法は鍵に対する全数探索であり，その全数探索を困難にさせるようなアルゴリズムは有用であると考えられる．

本論文では，Rivest の示した All-or nothing transform を満たす package transform を基に，悉無律を満たす暗号利用モードの設計と評価を行った．今回はプログラムのコード上，ECB モードと CBC モードのみの実験と考察を行った．それぞれのモードやブロック数の違いについて結果を得ることが出来た．

今後の課題として，package transform との実行時間などの比較や，その他の暗号利用モードでのプログラムの実行などが挙げられる．

謝辞 本研究を進めるにあたり，ご多忙にもかかわらず熱心にご指導頂きました，櫻井幸一教授に心から感謝致します．また, 研究に協力頂きました櫻井研究室の皆様に心から感謝致します．

参考文献

[Adi79] Shamir Adi. How to share a secret. Communications of the ACM, 22 (11): 612–613, doi:10.1145/359168.359176, 1979.
[IPA03] 独立行政情報処理推進機構. 「ブロック暗号を使った秘匿、メッセージ認証、及び認証暗号を目的とした利用モードの技術報告書」. CRYPTREC Report 2003, https://www.cryptrec.go.jp/exreport/cryptrec-ex-0205-2003.pdf(最終閲覧日 2021 年 1 月 30 日)
[Kur10] 黒澤 馨. 現代暗号への招待. サイエンス社, 2010.
[Kra93] Hugo Krawczyk. Secret Sharing Made Short. Annual International Cryptology Conference CRYPTO 1993: Advances in Cryptology — CRYPTO’ 93 pp 136-146,1993.
[KM18] Katarzyna Kapusta and Gerard Memmi. PE-AONT: Partial Encryption combined with an All-or-Nothing Transform. arXiv perprint arXiv:1811.09144, 2018.
[KO04] 黒澤 馨・尾形 わかは. 現代暗号の基礎数理. コロナ社, 2004.

n:ブロック数　s:サイズ

変換	n:10 s:16384k	n:100 s:1638k	n:1000 s:163.9k
1	12043	13846	20767
2	12274	15540	20725
3	11609	15496	20984
4	11999	15896	21672
5	12861	14074	21487
6	10763	15844	21410
7	12785	15421	21869
8	12872	14408	21090
9	11667	14906	20680
10	12766	14697	20783
average	12163.9	15012.8	21146.7

(ms)

逆変換	n:10 s:16384k	n:100 s:1638k	n:1000 s:163.9k
1	8739	8794	9899
2	8593	9091	9814
3	8750	9367	9925
4	8696	8661	9824
5	8639	8798	9825
6	8488	9078	9892
7	9094	9393	9979
8	9082	9514	9782
9	8698	9115	9824
10	8639	8834	9826
average	8741.8	9064.5	9859

(ms)

図 11　ECB モードの結果

Fig. 11　Result of ECB mode.

n:ブロック数　s:サイズ

変換	n:10 s:16384k	n:100 s:1638k	n:1000 s:163.9k
1	10719	14479	22638
2	10373	14542	23085
3	9773	15355	23228
4	10212	15324	22002
5	10334	15957	22026
6	10482	15744	22298
7	10330	15693	22263
8	10037	15066	22120
9	10230	14721	22427
10	10677	15602	22375
average	10316.7	15248.3	22446.2

(ms)

逆変換	n:10 s:16384k	n:100 s:1638k	n:1000 s:163.9k
1	9024	9110	9423
2	8710	9953	9102
3	8781	9705	9053
4	9004	9648	9006
5	9182	9615	8966
6	9319	9555	9085
7	8986	10149	8947
8	8939	9415	9426
9	9008	9384	8987
10	8993	9461	8929
average	8994.6	9599.5	9092.4

(ms)

図 12　CBC モードの結果

Fig. 12　Result of CBC mode.

[MM06]　Robert P. McEvoy and Colin C. Murphy. EFFICIENT ALL-OR-NOTHING ENCRYPTION USING CTR MODE. Proceedings of the International Conference on Security and Cryptography, 2006.

[NSG14]　https://github.com/nsg21/aont 2014 年 7 月 13 日更新 (最終閲覧日 2021 年 1 月 30 日)

[Rab90]　Michael O. Rabin. The Information Dispersal Algorithm and its Applications. In: Capocelli R.M. (eds) Sequences. Springer, New York, 1990.

[Ron97]　Rivest Ronald. All-or-nothing encryption and the package transform. FAST SOFTWARE ENCRYPTION Proceedings. Lecture Notes in Computer Science. 1267. pp. 210–218, 1997

[RP11]　Jason K. Reash and James S. Plank. AONT-RS: Blending Security and Performance in Dispersed Storage Systems. FIST'11, 2011.

[SO17]　Fatty Salem and Mohamed Osman. RAON-RB: A Verifiable Randomized Non-Separable Encryption Scheme for Secure Cloud Storage. International Journal of Computer Applications (0975 – 8887) Volume 179 – No.6, 2017.