形式化数学記述言語Mizarによる共通鍵暗号AESの形式化

信州大学大学院 工学系研究科 情報工学専攻

今村充志

研究背景

- 通信経路の安全性を確保することは必要不可欠
- 暗号プロトコルを利用

- 暗号プロトコルなどの安全性
  →判断には専門知識が必要

研究背景

- ISO29128
  →形式検証による暗号プロトコルの安全性評価の国際規格

Mizarを安全性評価ツールに使うことが出来ないか

研究背景

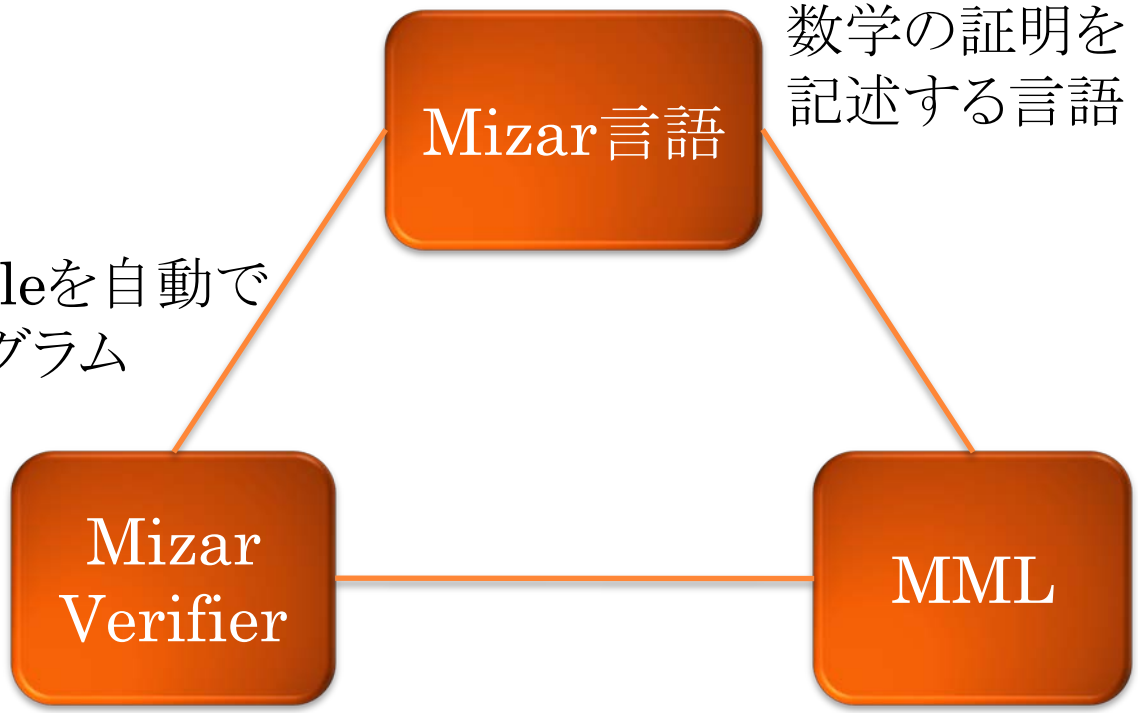
- Mizarを用いた安全性評価
  現在，利用可能なライブラリの拡充を行っている
  →確率，乱数生成器，ハッシュ関数，DES，AES etc

- 安全性評価ツールとしてのMizarの地位を確立

Mizarプロジェクト

- ポーランドのTrybulec教授らによって1973年に開始
- QEDプロジェクト
- 信州大学も参加
- カナダや中国，日本の他大学とも連携

Mizarシステム



数学の証明を
記述する言語

記述したarticleを自動で
検証するプログラム

証明済みのarticleを
集めたライブラリ

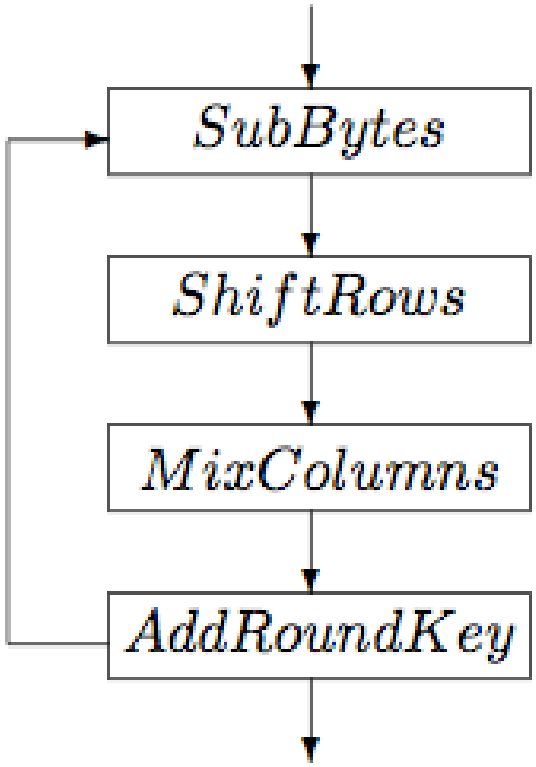
Advanced Encryption Standard

- 2001年にアメリカ国立標準局によって連邦情報処理標準として採択
- 共通鍵暗号
- ブロック長128ビット
- 鍵長128ビット，192ビット，256ビット

Advanced Encryption Standard

Advanced Encryption Standard

- ラウンド処理



AESの形式化

- 平文・秘密鍵を有限列の集合の要素と定義
- 暗号化アルゴリズムに出てくる操作，変換を各々定義
- 暗号化アルゴリズムを定義

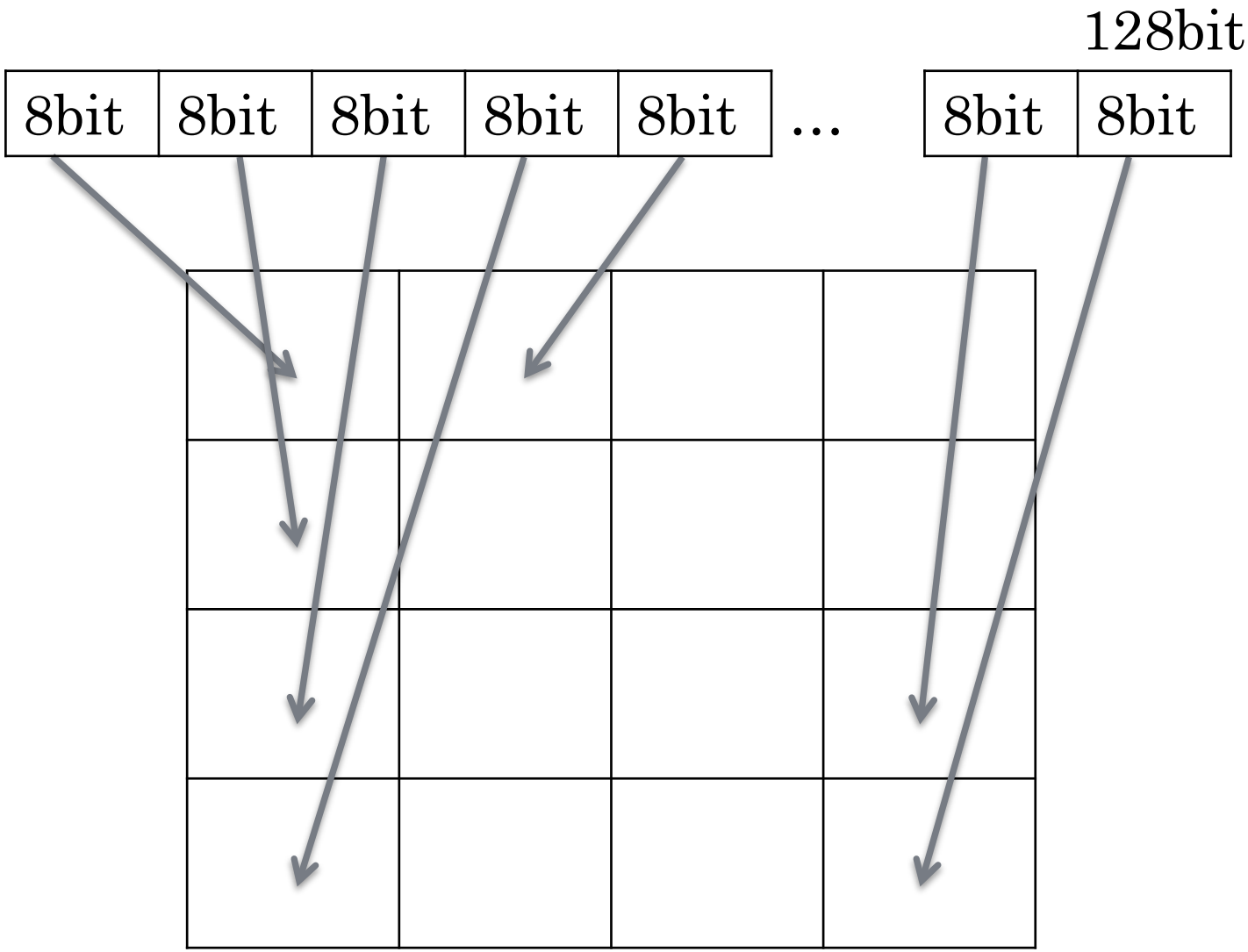
平文・秘密鍵の形式化

平文・秘密鍵はブロック長128ビットのデータ

→有限列の集合128-tuples_on BOOLEAN の要素

有限列 M	0 or 1	0 or 1	…	0 or 1	0 or 1
	M.1	M.2	…	M.127	M.128

テーブル化の形式化



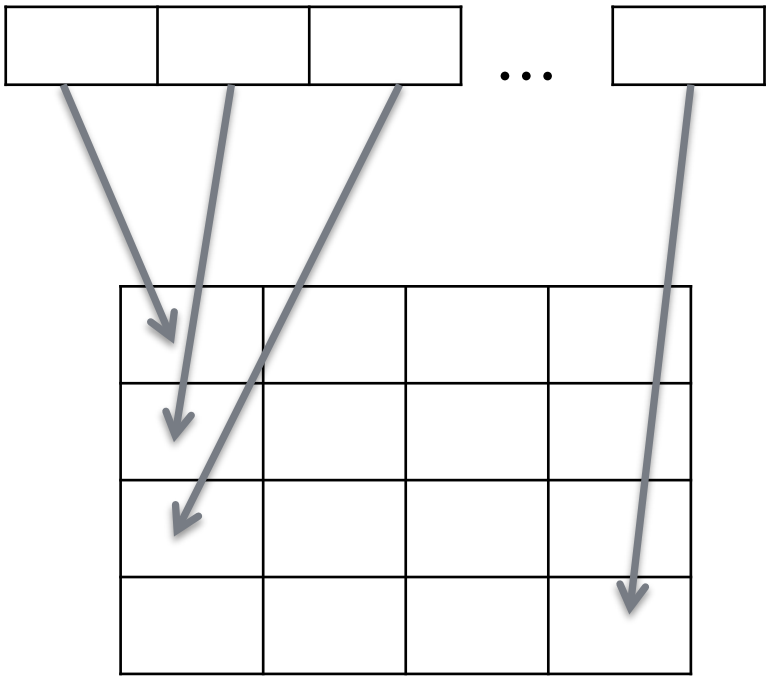
テーブル化の形式化

- テーブル化の入力と出力
  128-tuples_on BOOLEAN
  ↓
  4-tuples_on(4-tuples_on (8-tuples_on BOOLEAN))

- テーブル化したものをM
  一行一列目→(M.1).1
  二行一列目→(M.1).2

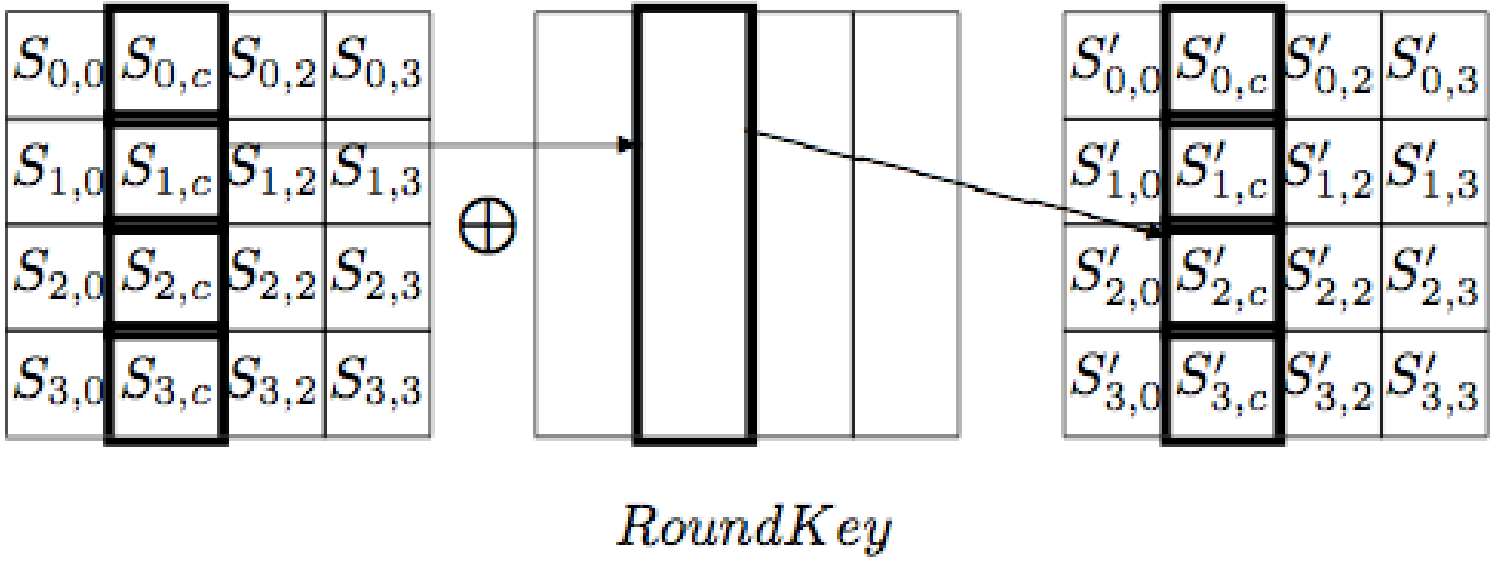
テーブル化の形式化

- Statearray
- ((it.input).i).j
  = mid (input,
  1+ (i -'1)*8 +
  (j -' 1)*32 ,
  1+ (i -'1)*8 +
  (j -' 1)*32+ 7))



- mid (input,1,8)
  →長さ8の有限列

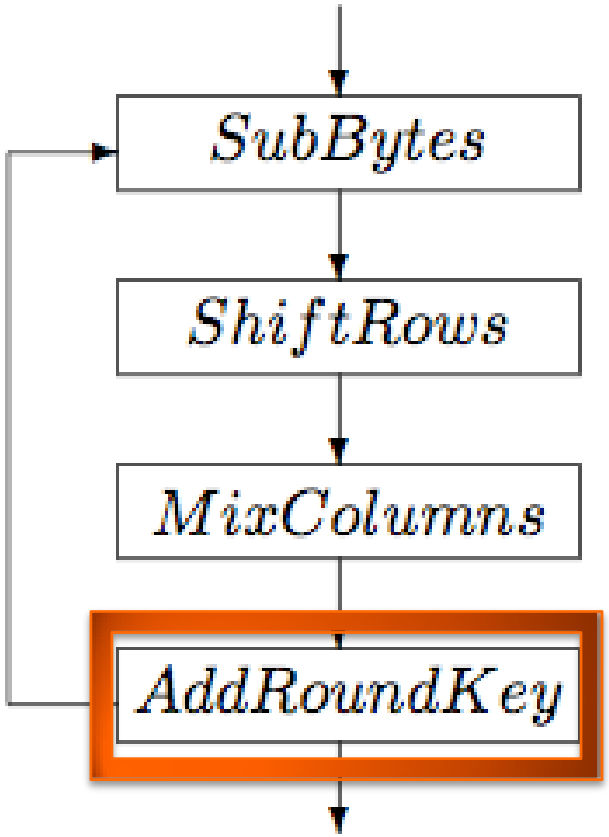
AddRoundKeyの形式化



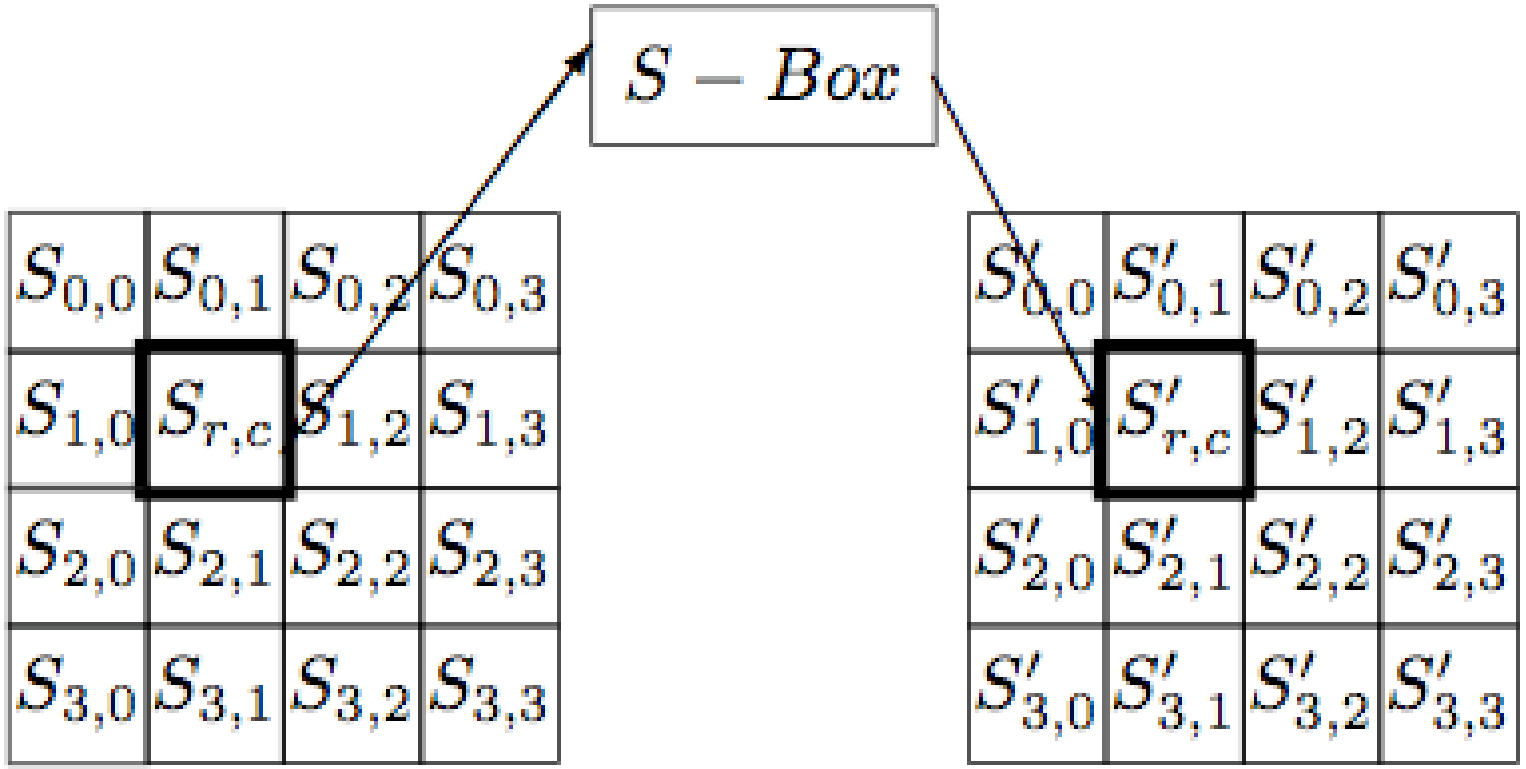
AddRoundKeyの形式化

- 入力
  4-tuples_on(4-tuples_on
  (8-tuples_on BOOLEAN))

- textij= (text.i).j &
  keyij= (key.i).j &
  ((it.(text,key)).i).j
  = Op-XOR(textij,keyij)



SubBytesの形式化



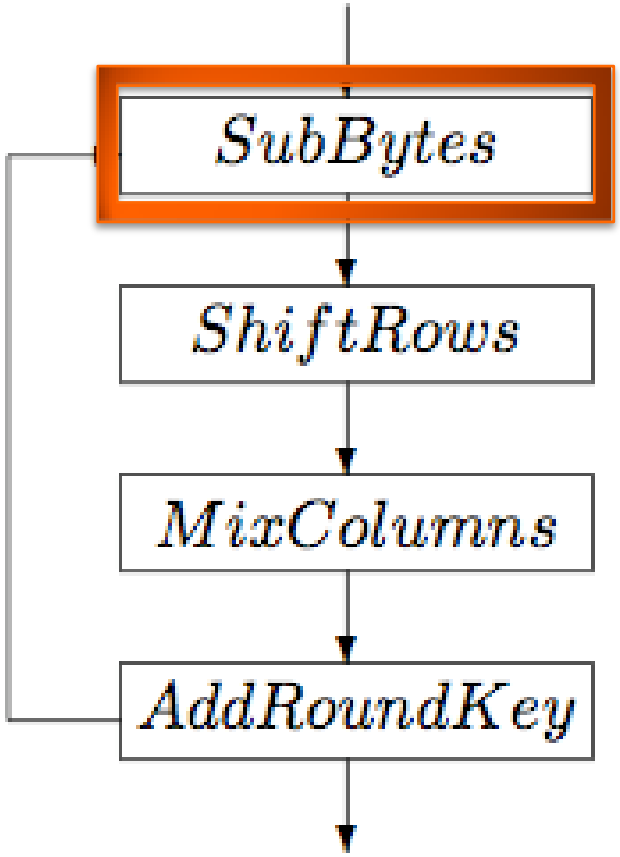
S-Box

		Y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
X	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	A	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	B	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	C	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	D	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	E	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	F	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

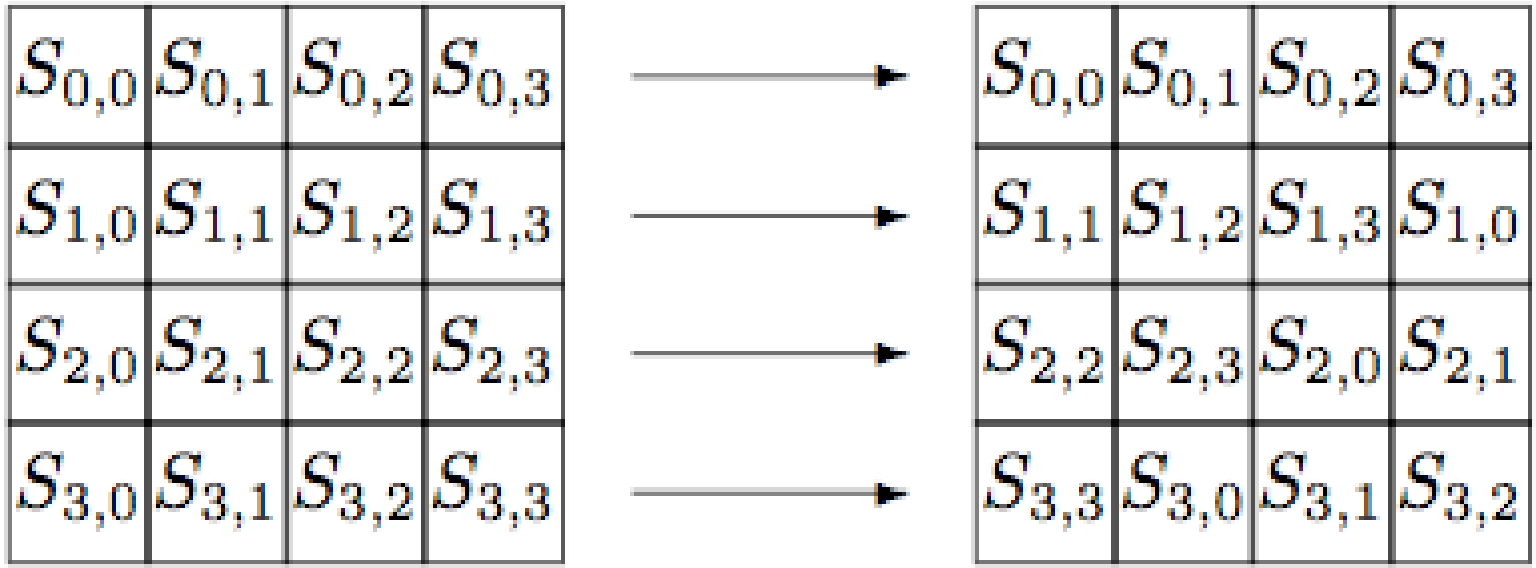
SubBytesの形式化

- 入力
  4-tuples_on(4-tuples_on
  (8-tuples_on BOOLEAN))

- inputij = (input.i).j &
  ((it.input).i).j = SBT.(inputij)



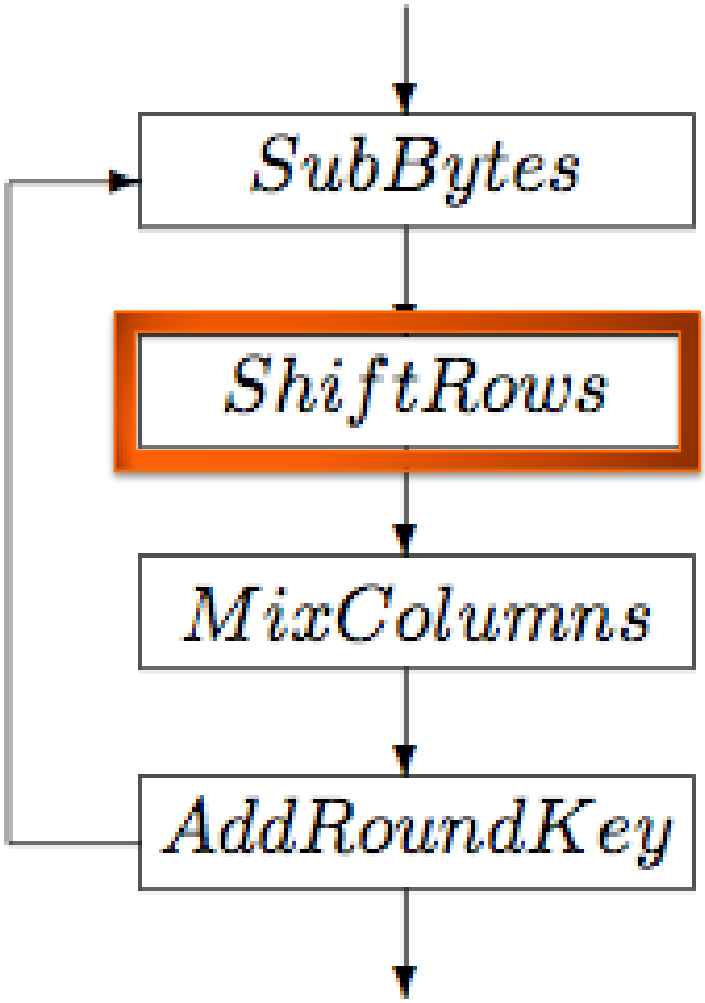
ShiftRowsの形式化



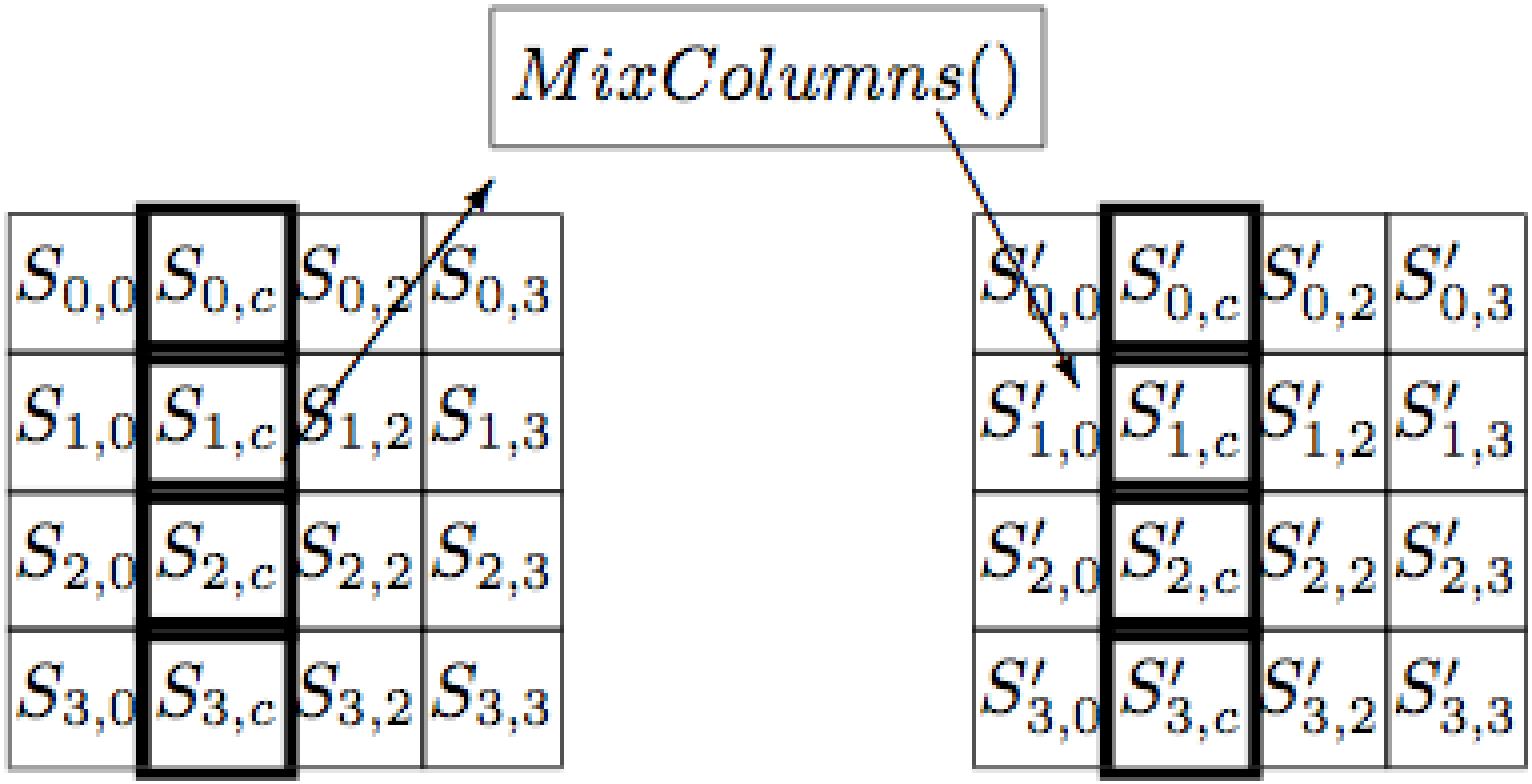
ShiftRowsの形式化

- 入力
  4-tuples_on(4-tuples_on
  (8-tuples_on BOOLEAN))

- xi = input.i &
  (it.input).i =
  Op-Shift(xi,5-i))



MixColumnsの形式化



MixColumnsの形式化

- GF(2^8)上において

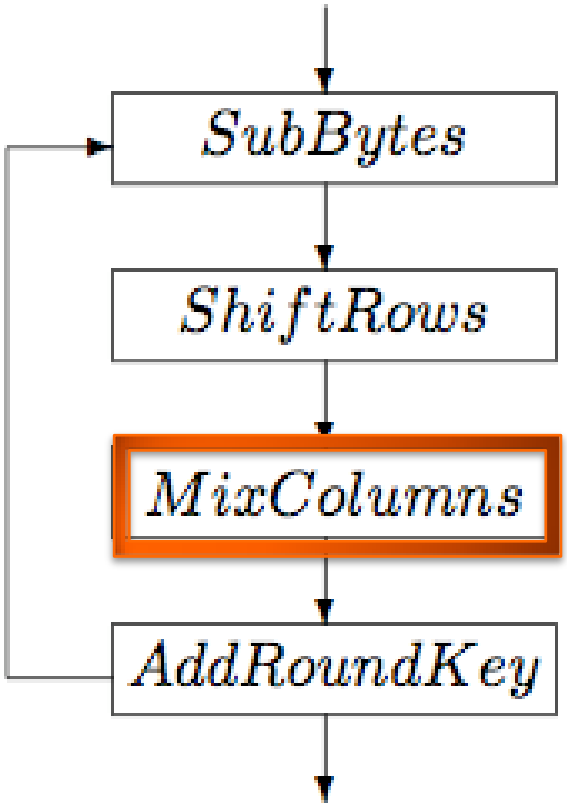
$$\begin{vmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \end{vmatrix} = \begin{vmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{vmatrix} \begin{vmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{vmatrix}$$

の行列式の計算を行う

MixColumnsの形式化

- 入力
  4-tuples_on(4-tuples_on
  (8-tuples_on BOOLEAN))

- x1 = (x.i).1&…&
  (y.1).i = Op-XOR (Op-XOR
  (Op-XOR (2 'gf' x1, 3 'gf' x2),
  1 'gf' x3),1 'gf' x4)
  &…&
  (y.4).i = Op-XOR (Op-XOR
  (Op-XOR (3 'gf' x1, 1 'gf' x2),
  1 'gf' x3),2 'gf' x4)



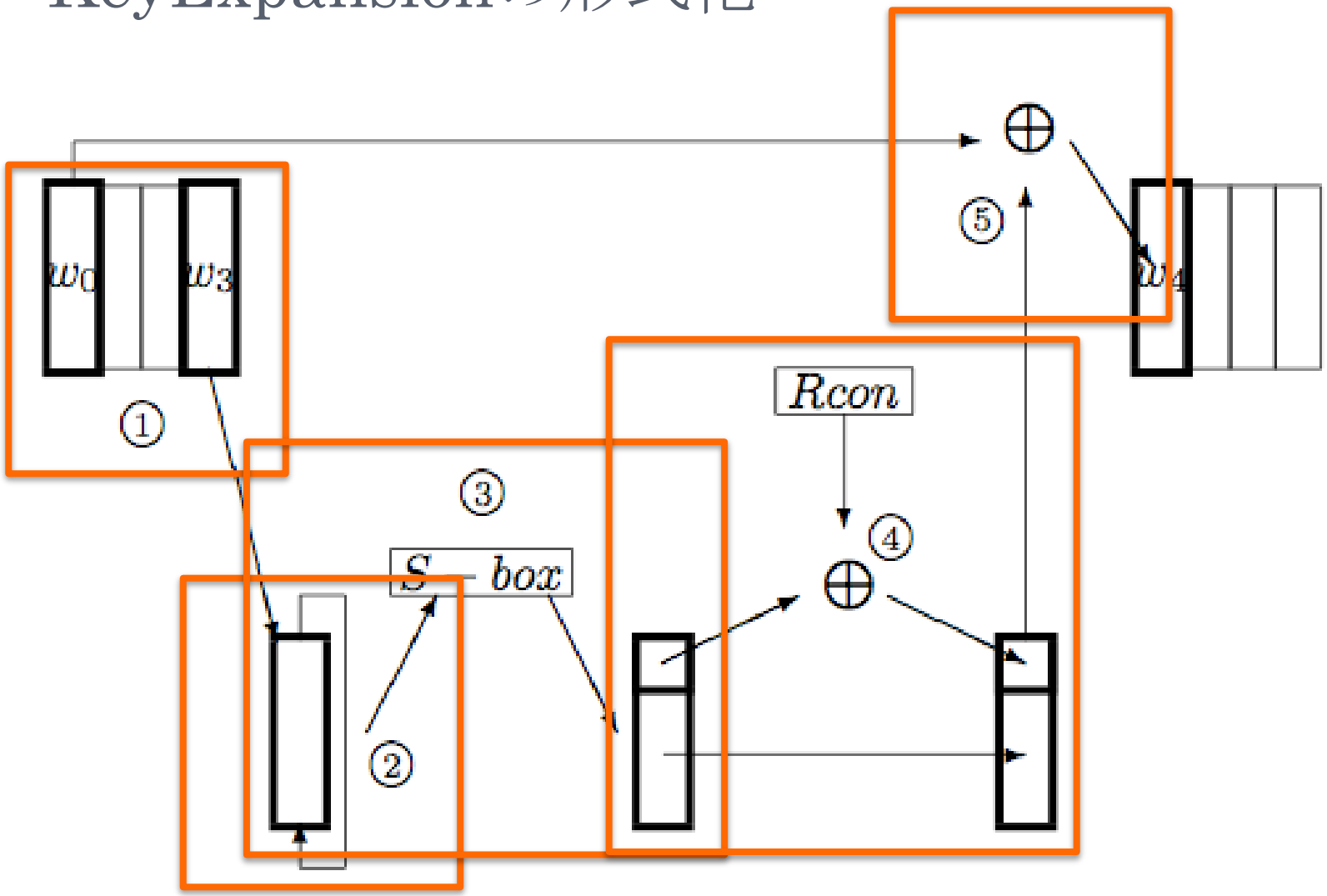
KeyExpansionの形式化

- テーブル化した4×4のラウンド鍵を
  $RoundKey_0$とする

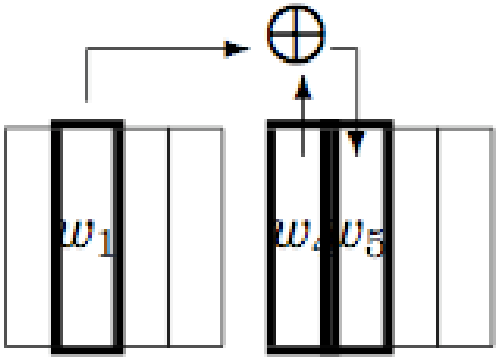
- $RoundKey_0$を用いて$RoundKey_1$…を生成

- 一列ごと生成
  →鍵長が128ビットの場合は4の倍数の列,
  　192ビットの場合6の倍数の列,
  　256ビットの場合8の倍数の列と
  それ以外の列の生成方法が異なる

KeyExpansionの形式化



KeyExpansionの形式化



n*4列目以外を求める場合

KeyExpansionの形式化

- KeyExpansion(SBT,m)

w = (KeyExpansionX(SBT,m)).Key &
for i be Nat st i <7+m holds
(it.Key).(i+1) = <*w.(4*i+1),w.(4*i+2),
w.(4*i+3),w.(4*i+4)*>

関数の一意性

- input,outputを
  Element of 4-tuples_on
  (4-tuples_on (8-tuples_on BOOLEAN))

- InvShiftRows.(ShiftRows.input) =input

暗号化関数AES-ENCの形式化

- textを4-tuples_on(4-tuples_on (8-tuples_on BOOLEAN))
- Keyをm-tuples_on(4-tuples_on (8-tuples_on BOOLEAN))

- AES-ENC(SBT,MixColumns,text,Key)
  = … &
  it=AddRoundKey.((ShiftRows*SubBytes(SBT)).(seq.(7+m-1)),KeyNr)

復号関数DES-DECの形式化

- textを4-tuples_on(4-tuples_on (8-tuples_on BOOLEAN))
- Keyをm-tuples_on(4-tuples_on (8-tuples_on BOOLEAN))

- AES-DEC(SBT,MixColumns,text,Key) =…& it=AddRoundKey.(seq.(7+m-1),KeyNr)

AESの一意復号性

- textを4-tuples_on(4-tuples_on (8-tuples_on BOOLEAN))
- Keyをm-tuples_on(4-tuples_on (8-tuples_on BOOLEAN))

- AES-DEC(SBT,MixColumns,AES-ENC(SBT,MixColumns,text,Key),Key) = text

まとめと今後の課題

- 共通鍵暗号方式AESをMizar言語を用いて形式化記述を行った

今後の課題

- 安全性評価ツールに必要なライブラリの拡充を進めていく

ご清聴ありがとうございました

- for i be Element of NAT st
  m <= i & i < 4*(7+m) holds
  ex P be Element of
  (4-tuples_on (8-tuples_on BOOLEAN)),
  Q be Element of
  4-tuples_on (8-tuples_on BOOLEAN)
  st
  P = (it.Key).((i-m)+1) & Q = (it.Key).i &
  (it.Key).(i+1)= Op-WXOR
  (P,KeyExTemp(SBT,m,i,Q)))

- (ex T3 be Element of (4-tuples_on (8-tuples_on BOOLEAN)) st

  T3 = Rcon.(i/m) & it=OpWXOR(SubWord(SBT,RotWord(w)), T3))

  if ((i mod m) = 0) ,

  (it= SubWord(SBT,w)) if (m=8 & (i mod 8) = 4)

  otherwise it = w;

```
ex seq be FinSequence of
          (4-tuples_on(4-tuples_on (8-tuples_on BOOLEAN)))
    st
      len seq = 7+m-1 &
      ( ex  Keyi1 be Element of
                4-tuples_on(4-tuples_on (8-tuples_on BOOLEAN))
          st  Keyi1 =  ((KeyExpansion(SBT,m)).(Key)).1 &
             seq.1 = AddRoundKey.(text,Keyi1))
      &  (for i be Nat st 1<=i & i <  7+m-1
          holds
           ex  Keyi be Element of
                4-tuples_on(4-tuples_on (8-tuples_on BOOLEAN))
           st Keyi = ((KeyExpansion(SBT,m)).(Key)).(i+1)
           & seq.(i+1)
            =
AddRoundKey.((MixColumns*ShiftRows*SubBytes(SBT)).(seq.i),Keyi)
        )
      &  ex  KeyNr be Element of
                4-tuples_on(4-tuples_on (8-tuples_on BOOLEAN))
       st KeyNr = ((KeyExpansion(SBT,m)).(Key)).(7+m) &
       it=AddRoundKey.((ShiftRows*SubBytes(SBT)).(seq.(7+m-1)),KeyNr);
```

- ex seq be FinSequence of
- (4-tuples_on(4-tuples_on (8-tuples_on BOOLEAN)))
- st
- len seq = 7+m-1 &
- (ex Keyi1 be Element of
- 4-tuples_on(4-tuples_on (8-tuples_on BOOLEAN))
- st Keyi1 = (Rev((KeyExpansion(SBT,m)).(Key))).1 &
- seq.1
- = (InvSubBytes(SBT)*InvShiftRows).(AddRoundKey.(text,Keyi1)))
- & (for i be Nat st 1<=i & i < 7+m-1
- holds
- ex Keyi be Element of
- 4-tuples_on(4-tuples_on (8-tuples_on BOOLEAN))
- st Keyi = (Rev((KeyExpansion(SBT,m)).(Key))).(i+1)
- & seq.(i+1)
- = (InvSubBytes(SBT)*InvShiftRows*(MixColumns")).
- (AddRoundKey.(seq.i,Keyi)))
- & ex KeyNr be Element of
- 4-tuples_on(4-tuples_on (8-tuples_on BOOLEAN))
- st KeyNr = (Rev((KeyExpansion(SBT,m)).(Key))).(7+m) &
- it=AddRoundKey.(seq.(7+m-1),KeyNr);