2014. 2.13
NICT 情報セキュリティ シンポジウム
@ コクヨホール

RC4の脆弱性とSSL/TLSへの攻撃

五十部　孝典
ソニー株式会社

本日の内容 (1/2)

2012年度 CRYPTREC技術報告書 “ストリーム暗号RC4の安全性評価”

-128 bit key RC4 (SSL3.0 /TLS 1.0以上)の安全性評価

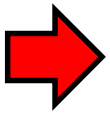
代表：五十部　孝典　(ソニー株式会社 / 神戸大学)

共同研究者：大東　俊博 (広島大学), 森井　昌克 (神戸大学)

- 神戸大学　学生：渡辺 優平, 長尾 篤, 塚畝 翼

http://www.cryptrec.go.jp/estimation/techrep_id2205.pdf

- RC4の既知の解析結果のサーベイ
- 新しい攻撃法の提案
  - 平文回復攻撃 [FSE 2013]
    - Broadcast setting
    - Multi-session setting (SSL/TLS)



2013年 RC4はCRYPTRECの推奨暗号リストから除外され，運用監視リストへ

本日の内容 (2/2)

その後の研究動向

- SSL/TLSへの平文回復攻撃の改良
  - 現実的な平文パターンでの評価 [ICSS 2013]
  - 比較的安全な実装方法(RC4-drop)への拡張 [SAC 2013]
  - 成功確率の向上 [USENIX 2013]

- WPA-TKIPへの攻撃の拡張
  - 平文回復攻撃 [FSE 2014]

発表の流れ

1. ストリーム暗号RC4

2. RC4の安全性

3. 新しい攻撃法：平文回復攻撃
   - Broadcast setting
   - Multiple session setting (SSL/TLS)

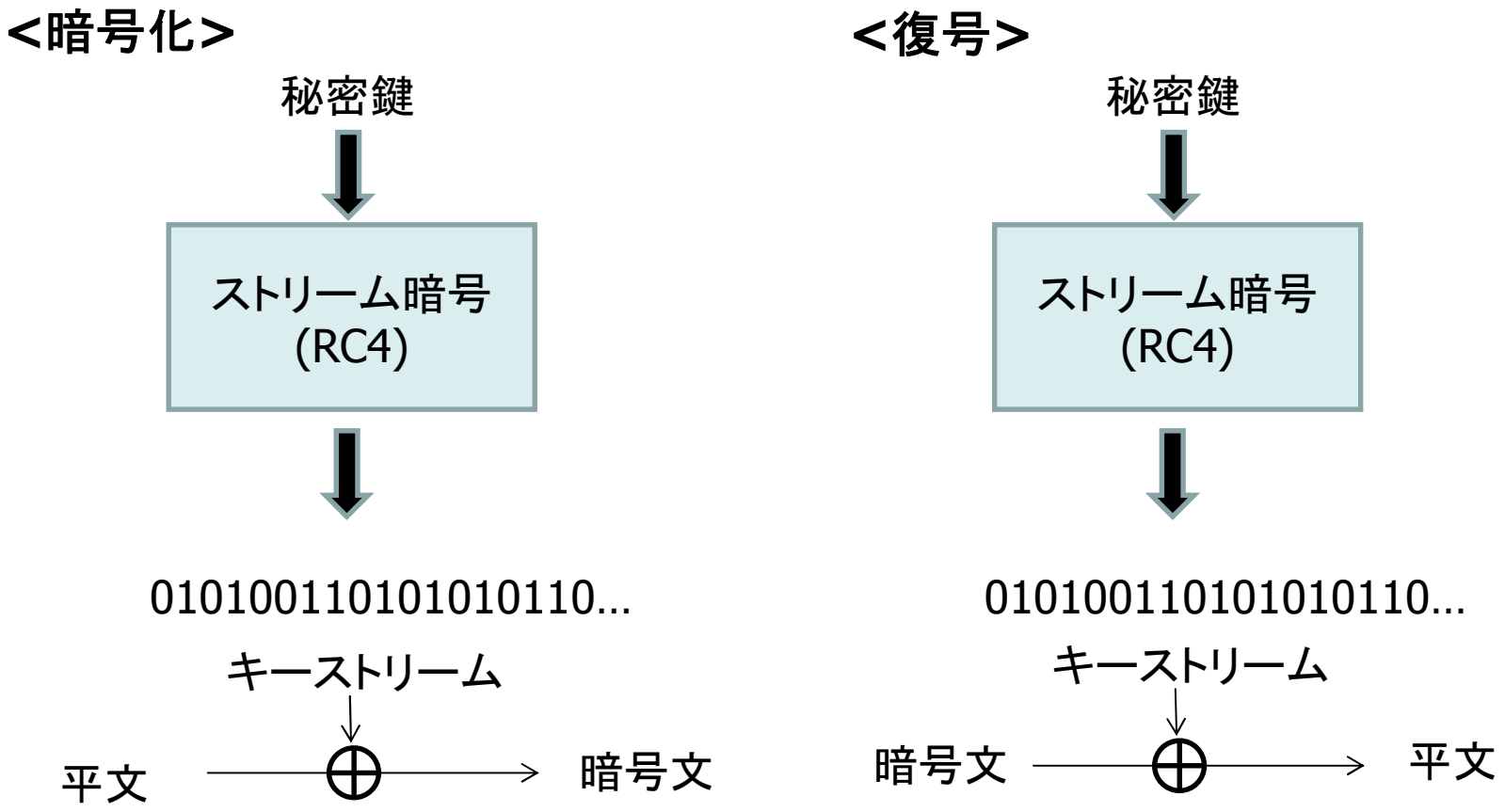
4. その後の進展

CRYPTREC技術報告書
に記載されている内容

1. ストリーム暗号RC4

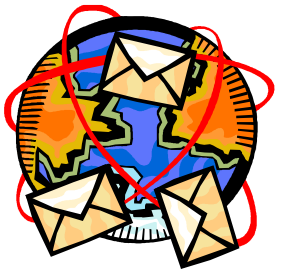
ストリーム暗号

- 共通鍵暗号のひとつ
  - 秘密鍵から擬似乱数系列(キーストリーム)を生成する関数

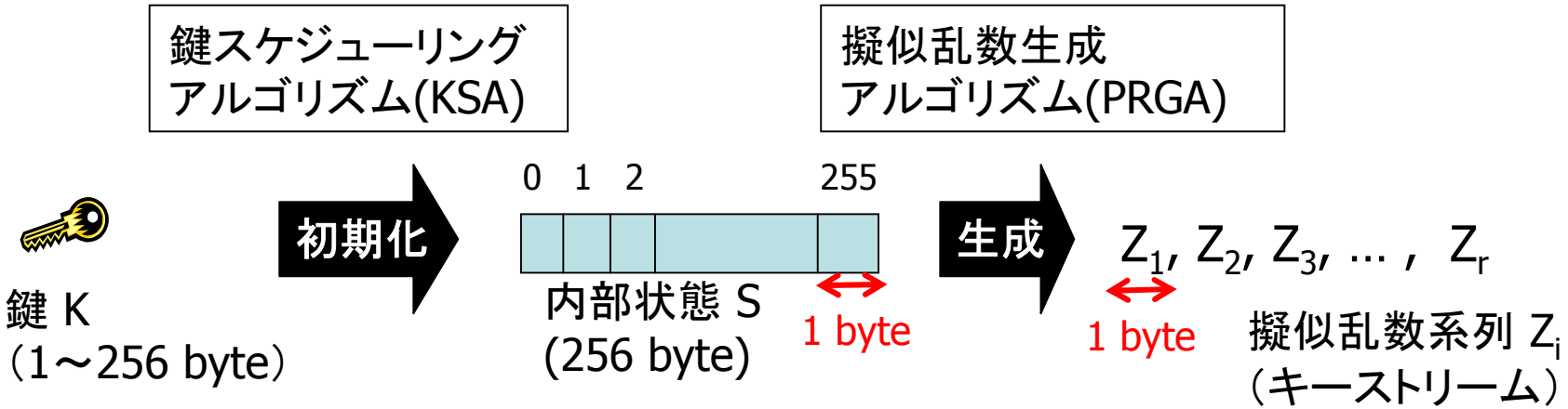


RC4

- 1987年にRivestにより開発されたストリーム暗号
  - 最も広く使われている暗号の一つ
    - WEP, WPA-TKIP, SSL/TLS, SSHなど

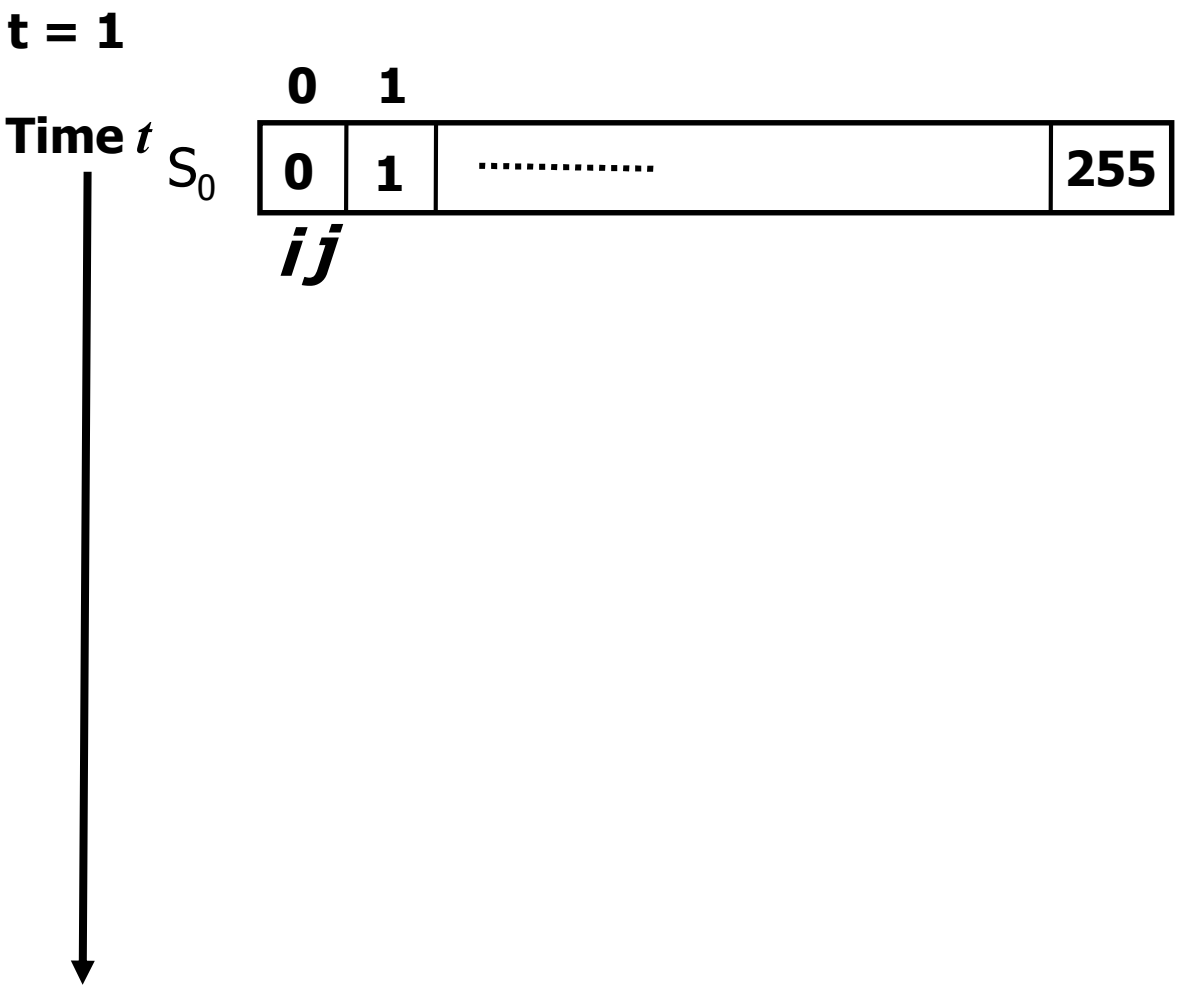


RC4の構造



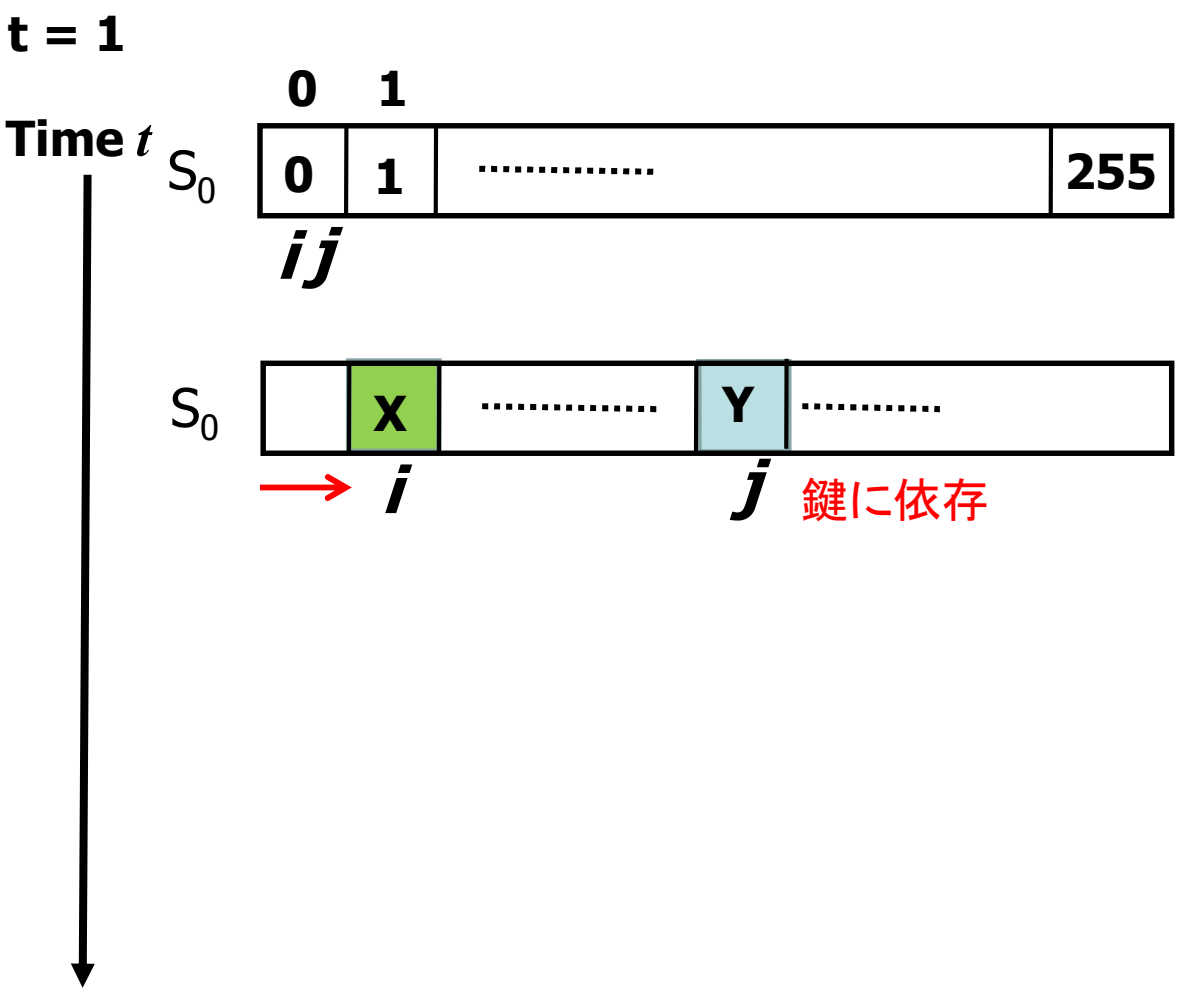
- バイト単位の処理
- 可変長鍵(1～256 byte)、推奨値は16 byte(128 bit)
- 内部状態は256 byteの配列S と 2つのindex i と j

鍵スケジューリングアルゴリズム(KSA)



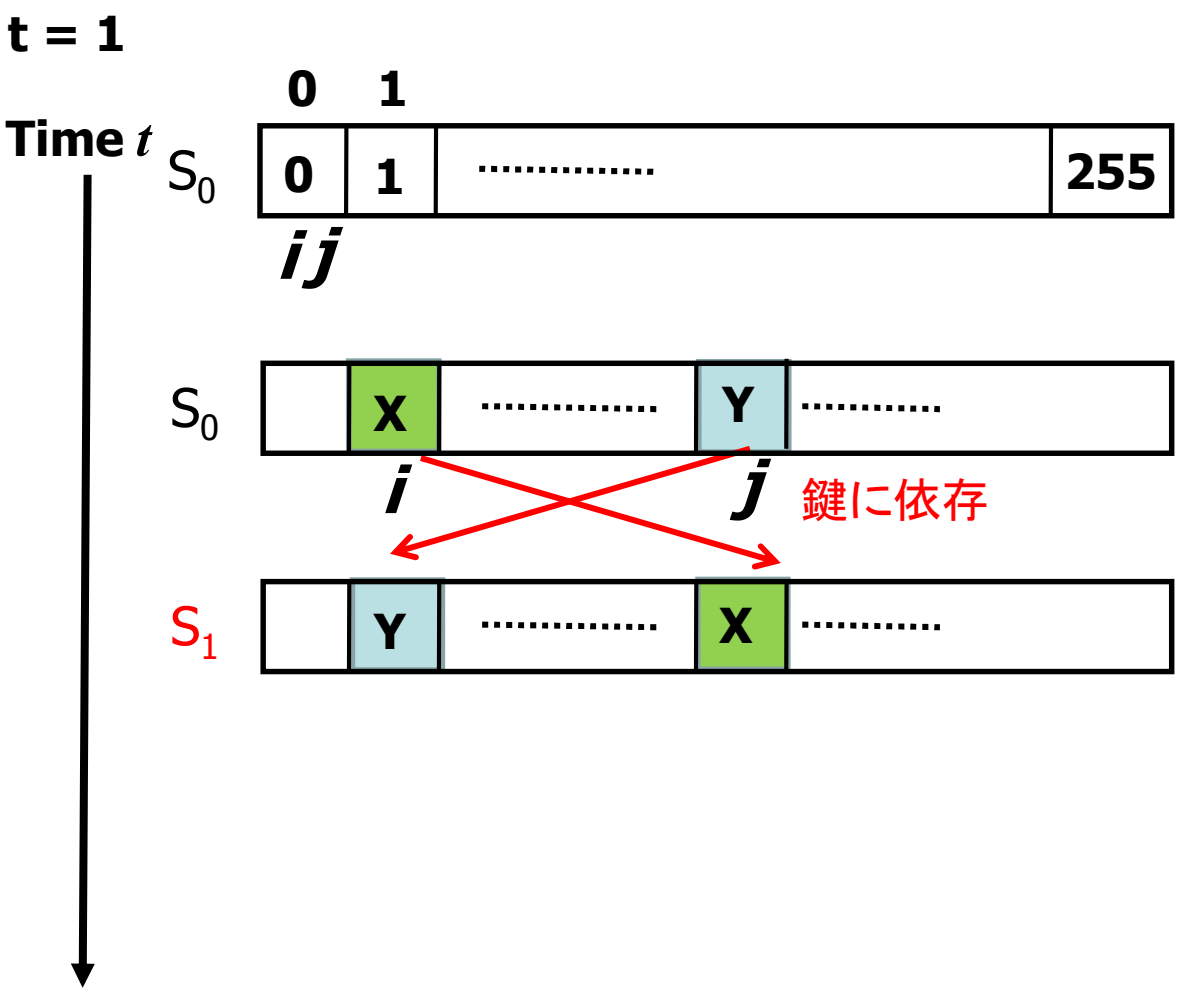

```
 i = 0
 J = 0
 S0[x] = x
loop
  j = j +S[ i ] + K[i]
  swap(S[ i ], S[ j ])
  i = i + 1
end loop
```

鍵スケジューリングアルゴリズム(KSA)



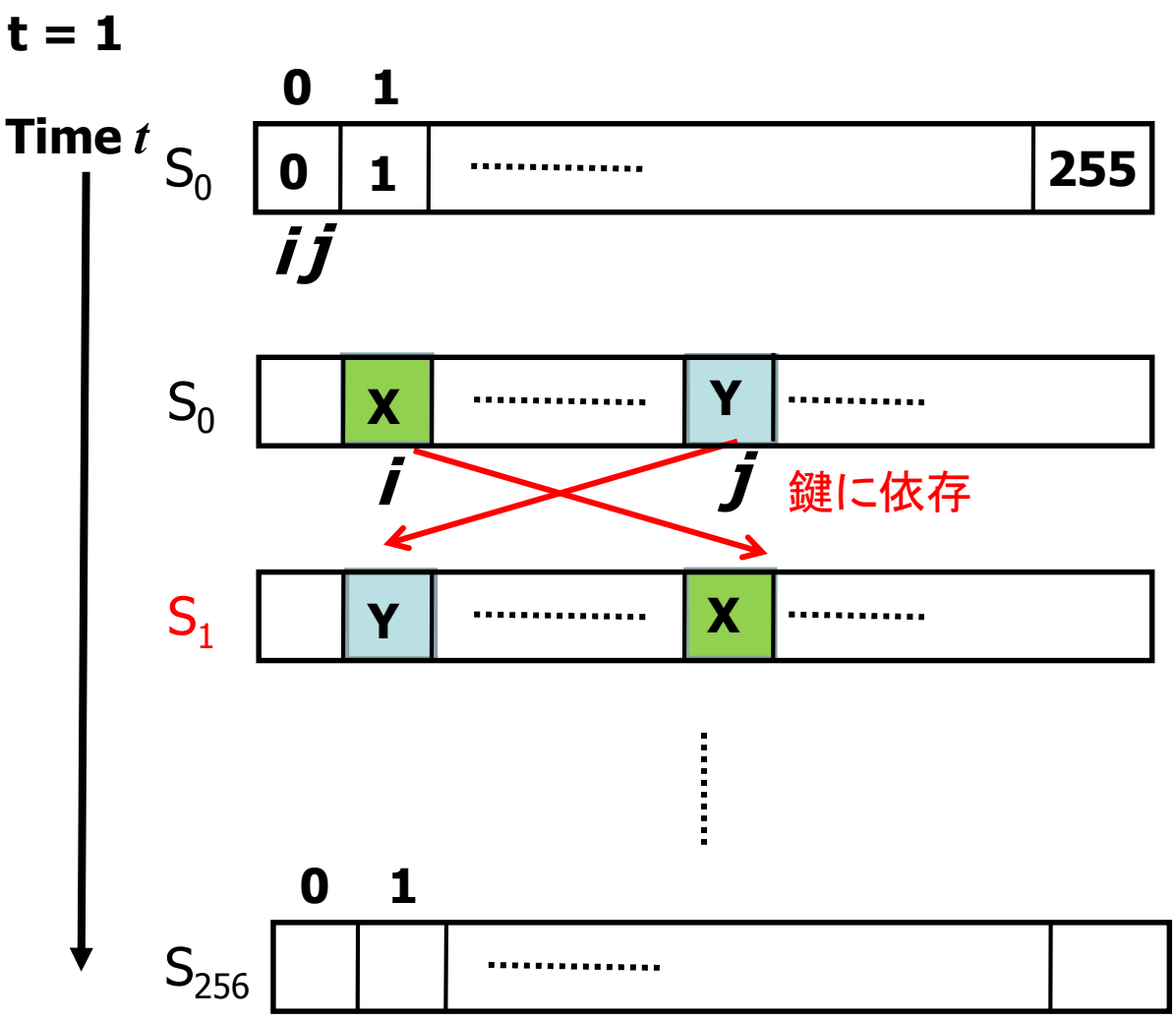

```
 i = 0
 J = 0
 S0[x] = x
loop
  j = j +S[ i ] + K[i]
  swap(S[ i ], S[ j ])
  i = i + 1
end loop
```

鍵スケジューリングアルゴリズム(KSA)



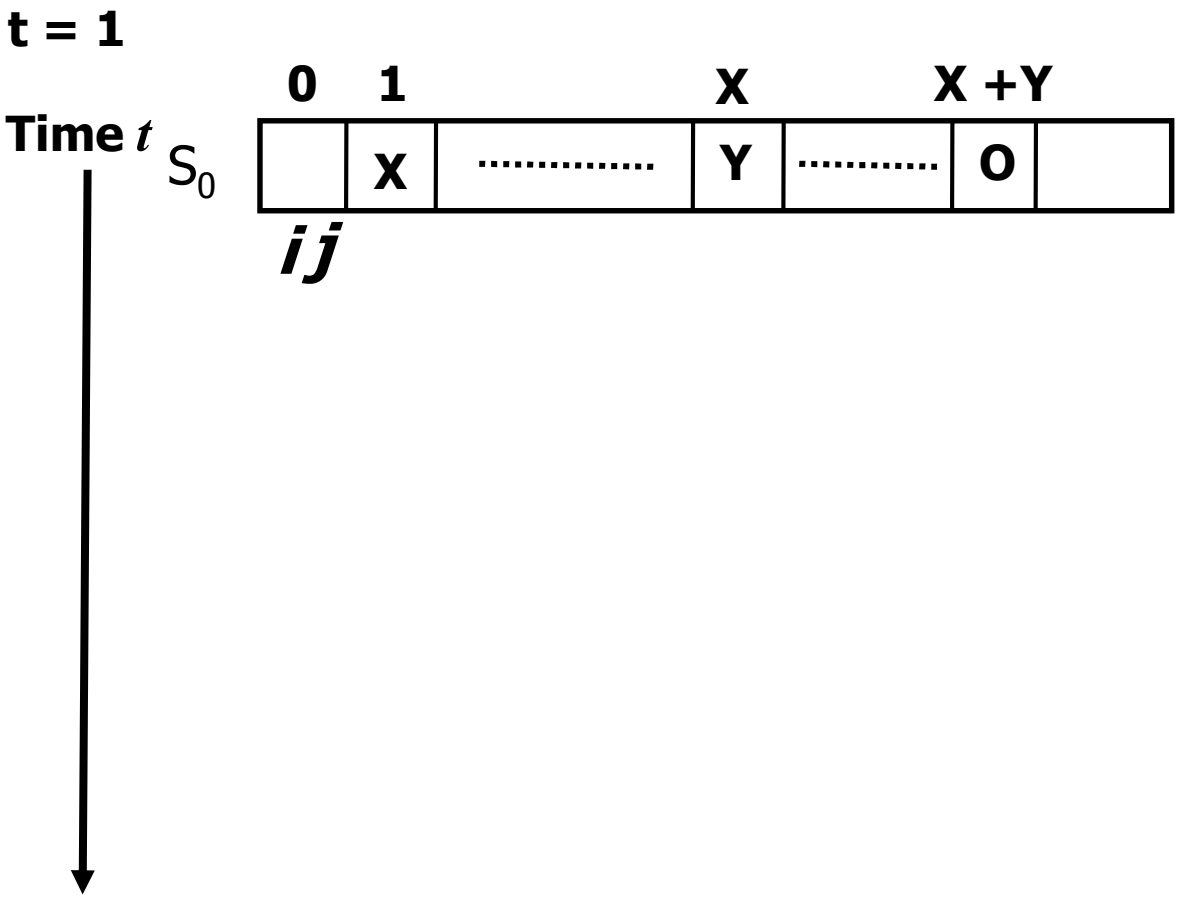

```
 i = 0
 J = 0
 S0[x] = x
loop
  j = j +S[ i ] + K[i]
  swap(S[ i ], S[ j ])
  i = i + 1
end loop
```

鍵スケジューリングアルゴリズム(KSA)



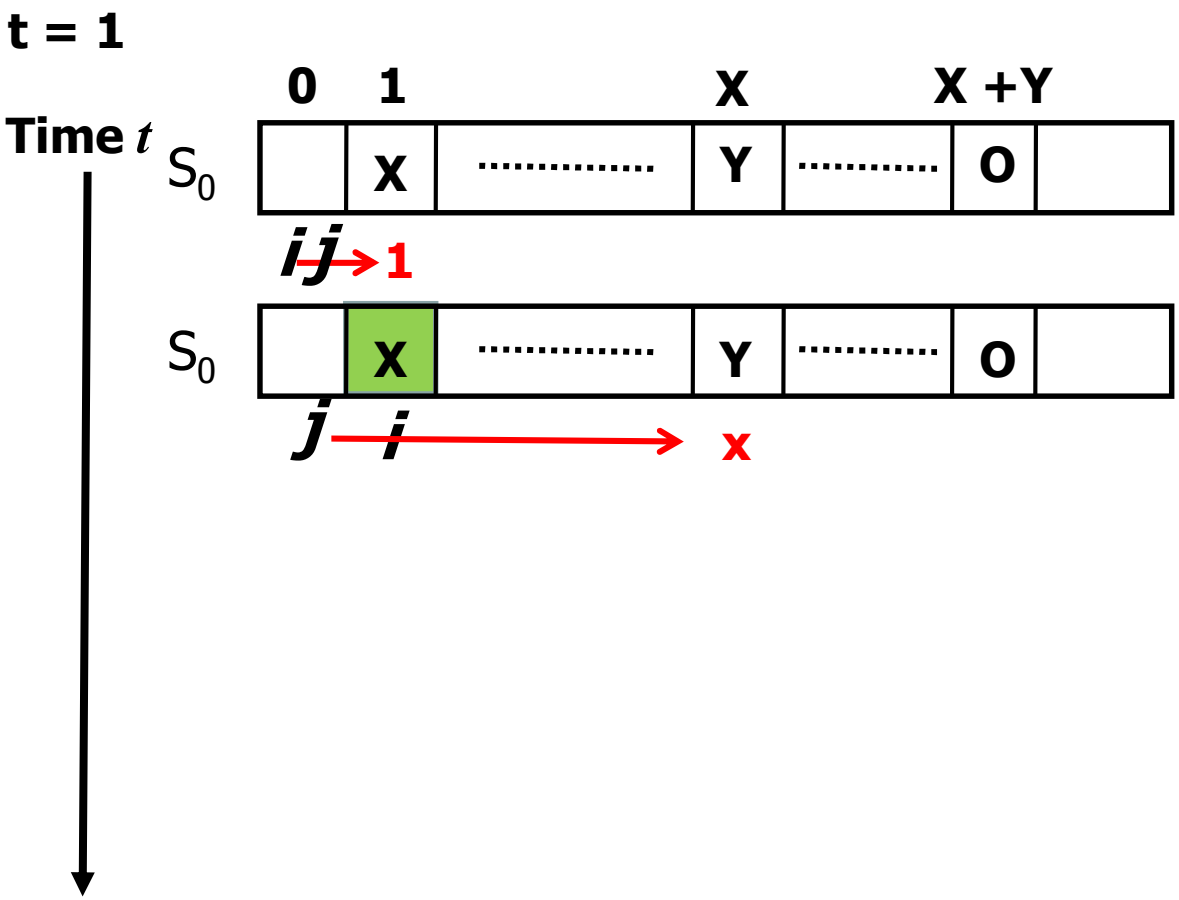

```
 i = 0
 J = 0
 S0[x] = x
loop
  j = j +S[ i ] + K[i]
  swap(S[ i ], S[ j ])
  i = i + 1
end loop
```

擬似乱数生成アルゴリズム (PRGA)



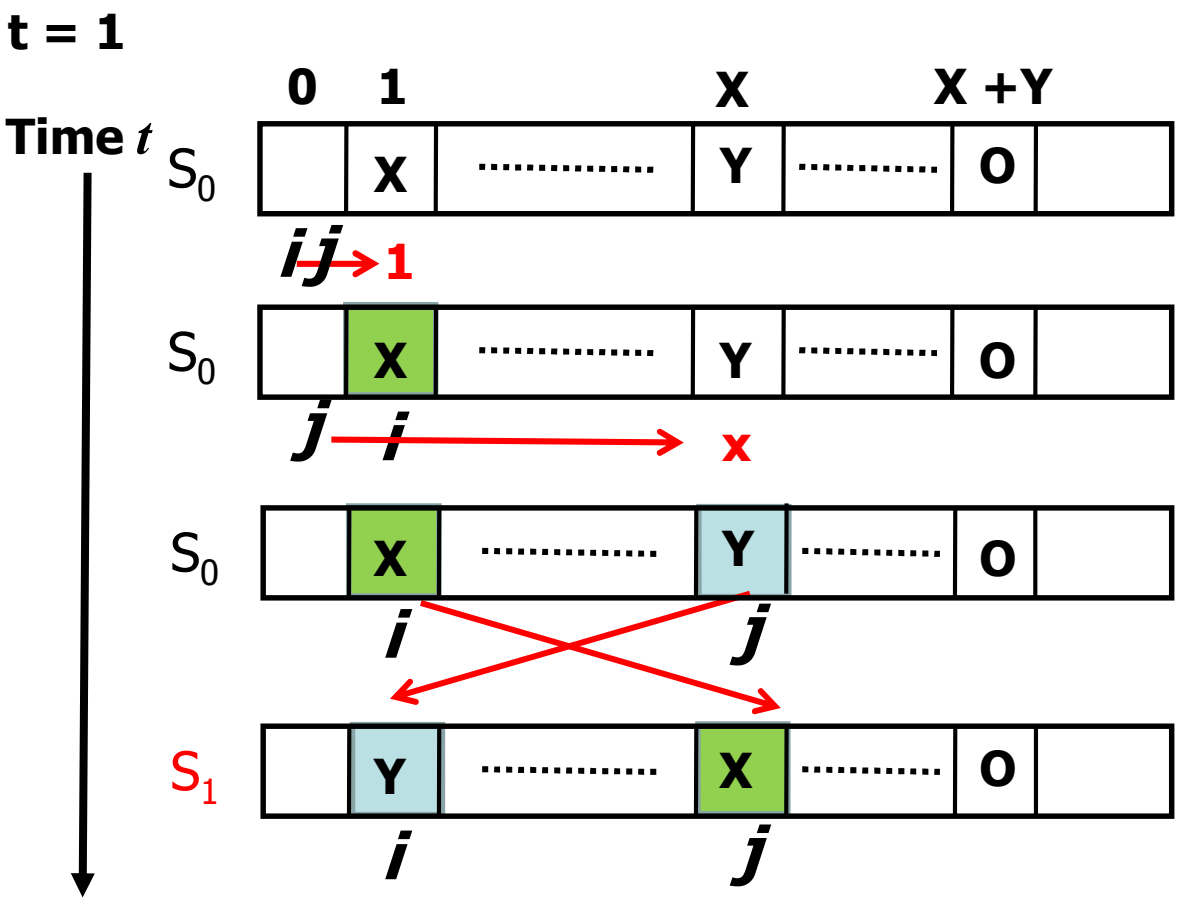

```
i = 0
J = 0
Loop
  i = i + 1
  j = j +S[ i ]
  swap(S[ i ], S[ j ])
  Z = S[S[ i ]+S[ j ]]
end loop
```

擬似乱数生成アルゴリズム (PRGA)



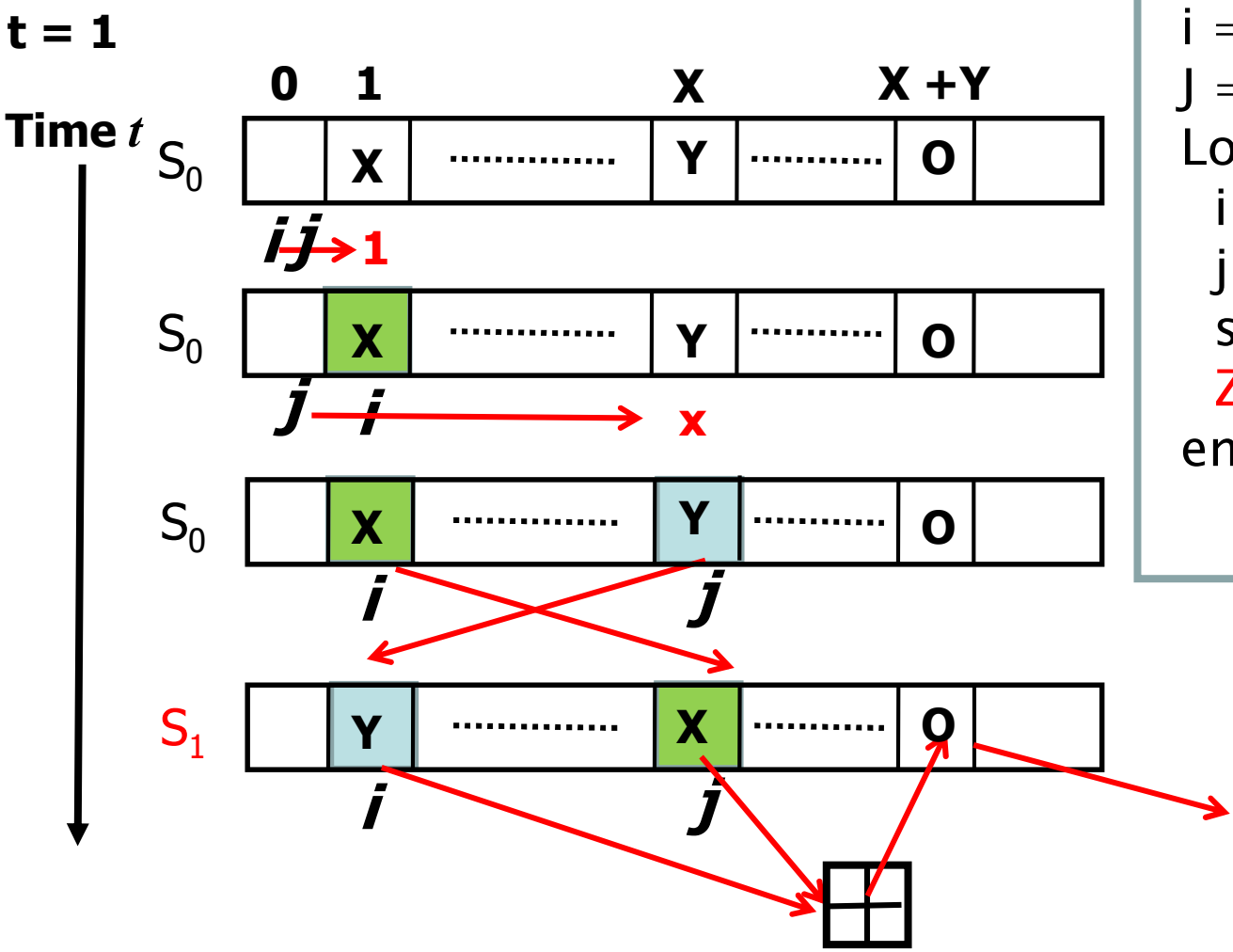

```
i = 0
J = 0
Loop
  i = i + 1
  j = j +S[ i ]
  swap(S[ i ], S[ j ])
  Z = S[S[ i ]+S[ j ]]
end loop
```

擬似乱数生成アルゴリズム (PRGA)




```
i = 0
J = 0
Loop
  i = i + 1
  j = j +S[ i ]
  swap(S[ i ], S[ j ])
  Z = S[S[ i ]+S[ j ]]
end loop
```

擬似乱数生成アルゴリズム (PRGA)

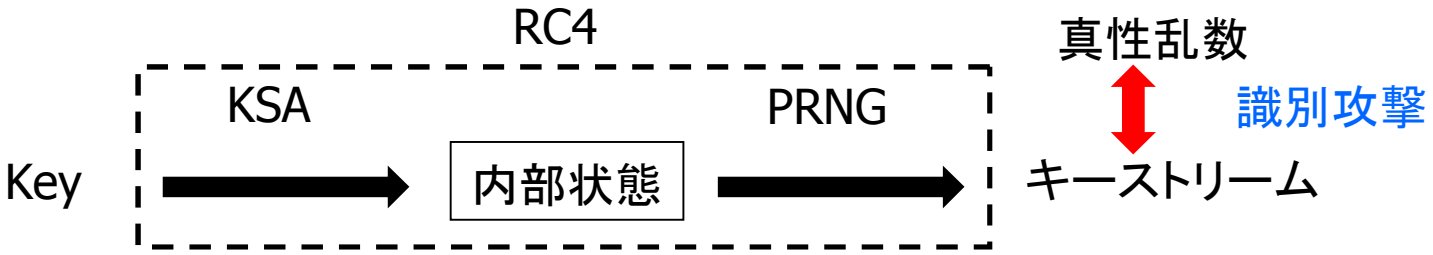



```
i = 0
J = 0
Loop
  i = i + 1
  j = j +S[ i ]
  swap(S[ i ], S[ j ])
  Z = S[S[ i ]+S[ j ]]
end loop
```

2. RC4の安全性

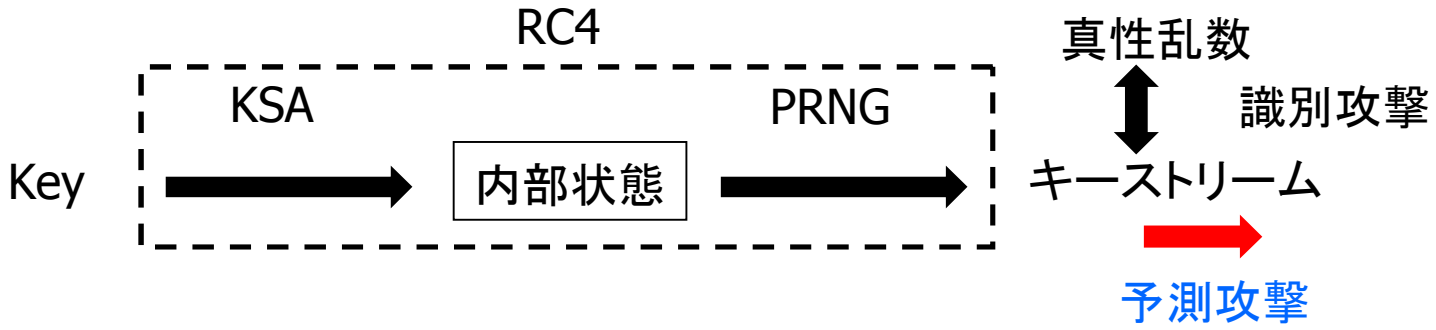
ストリーム暗号に求められる代表的な安全性

- 出力系列の乱数性
  - 出力系列(キーストリーム)を真性乱数と識別することが困難



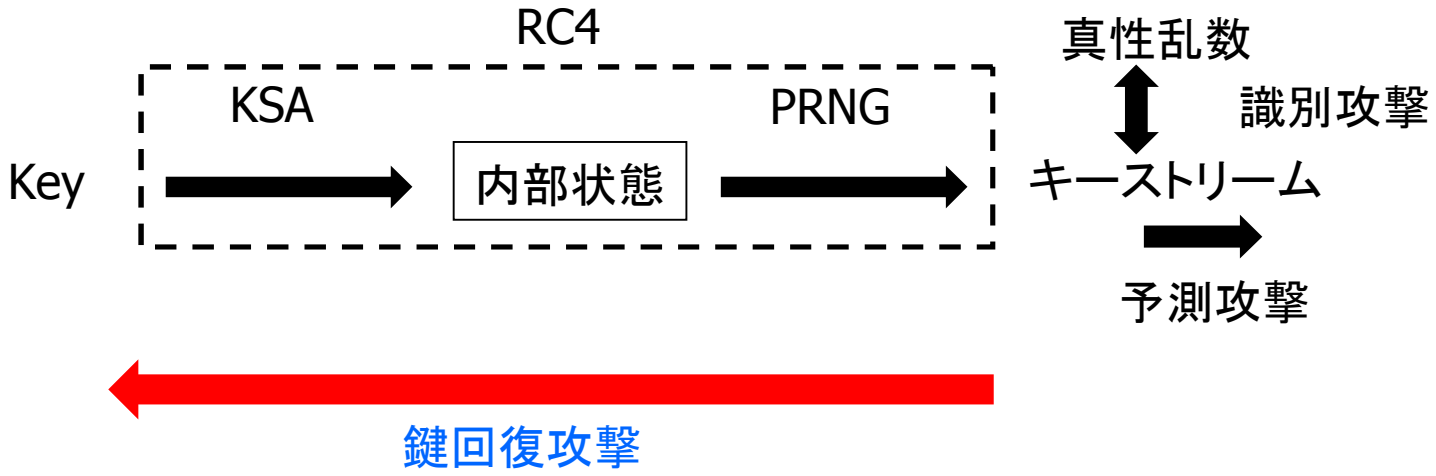
ストリーム暗号に求められる代表的な安全性

- 出力系列の乱数性
  - 出力系列(キーストリーム)を真性乱数と識別することが困難
- 出力の予測困難性
  - あるキーストリーム系列の集合から，以降のキーストリームの予測が困難



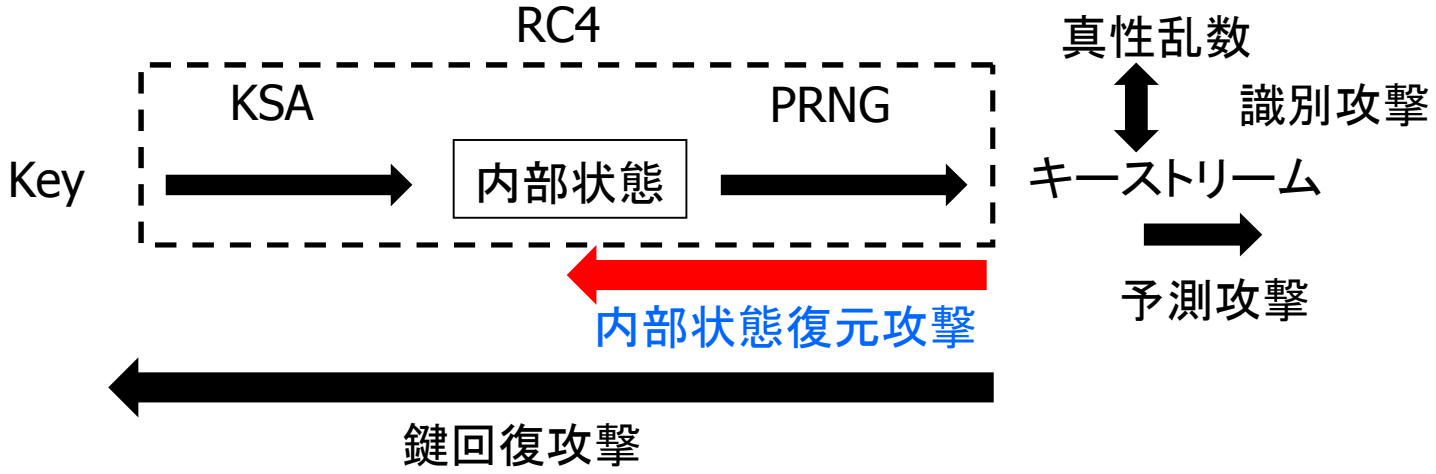
ストリーム暗号に求められる代表的な安全性

- 出力系列の乱数性
  - 出力系列(キーストリーム)を真性乱数と識別することが困難
- 出力の予測困難性
  - あるキーストリーム系列の集合から，以降のキーストリームの予測が困難
- 秘密鍵回復困難性
  - キーストリームから秘密鍵を求めることが困難

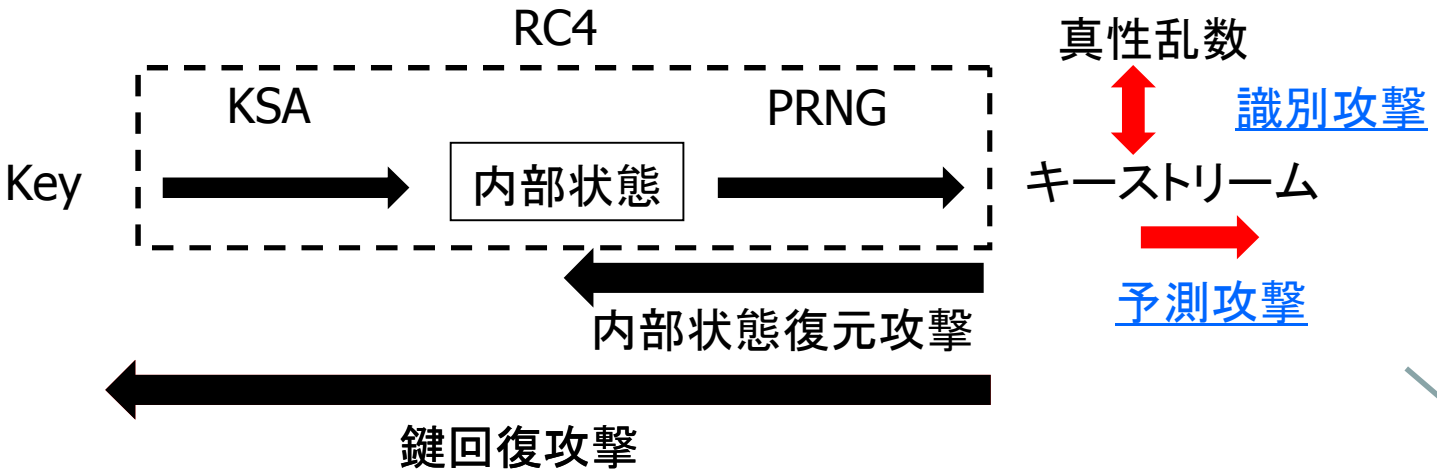


ストリーム暗号に求められる代表的な安全性

- 出力系列の乱数性
  - 出力系列(キーストリーム)を真性乱数と識別することが困難
- 出力の予測困難性
  - あるキーストリーム系列の集合から，以降のキーストリームの予測が困難
- 秘密鍵回復困難性
  - キーストリームから秘密鍵を求めることが困難
- 内部状態復元困難性
  - キーストリームから内部状態を復元することが困難



RC4の既知の安全性評価結果



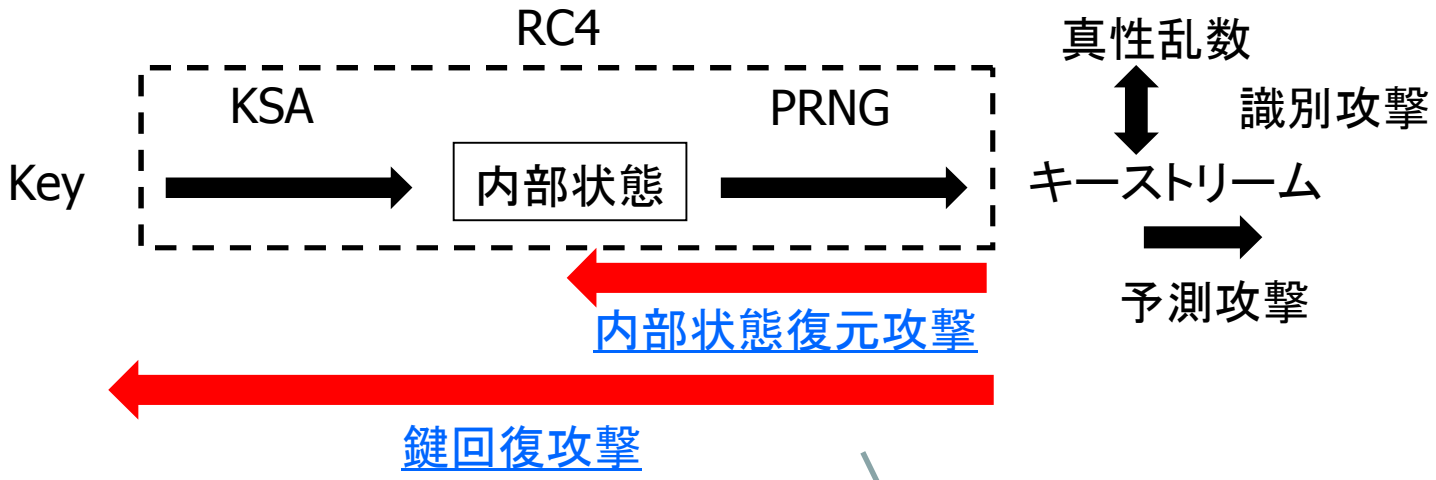
予測攻撃

Mantin [EUROCRYPT 2005]
: 2^{45} バイトのキーストリーム
から85%の確率で
1ビットの予測が可能

識別攻撃

Golic [EUROCRYPT 1997]:
: $2^{44.7}$ byteのキーストリームにより識別可能
Fluhrer et.al [FSE 2000]
: $2^{30.6}$ byteのキーストリームにより識別可能
Mantin [EUROCRYPT 2005]
: $2^{26.5}$ byteのキーストリームにより識別可能
(Multiple key) Mantin, Shamir [FSE 2001]
: 2^{8} byteのキーストリームにより識別可能

RC4の既知の安全性評価結果



内部状態復元攻撃

Knudsen et.al [ASIACRYPT 1998]
: 計算量 2^{779}
白石，大東，森井 [IEICE 2003]
: 計算量 2^{612}
Miximov et.al [CRYPTO 2008]
: **計算量 2^{241}**

鍵長を241 bitより長くしても安全性は向上しない

鍵回復攻撃 (weak key)

Roos [R'1995]
: 計算量2^{112}, 確率$2^{-10.9}$
Sepehrdad et al. [SAC 2010]
: 計算量$2^{38.09}$, 確率$2^{-87.9}$
計算量1, 確率$2^{-122.06}$
Our [JIP 2014]
: **計算量$2^{96.36}$, 確率$2^{-18.75}$**

RC4の既知の安全性評価まとめ

- 理論的な安全性
  - 識別攻撃により，擬似乱数との識別が容易
  - 全数探索より効率的に鍵の探索ができるweak keyが存在

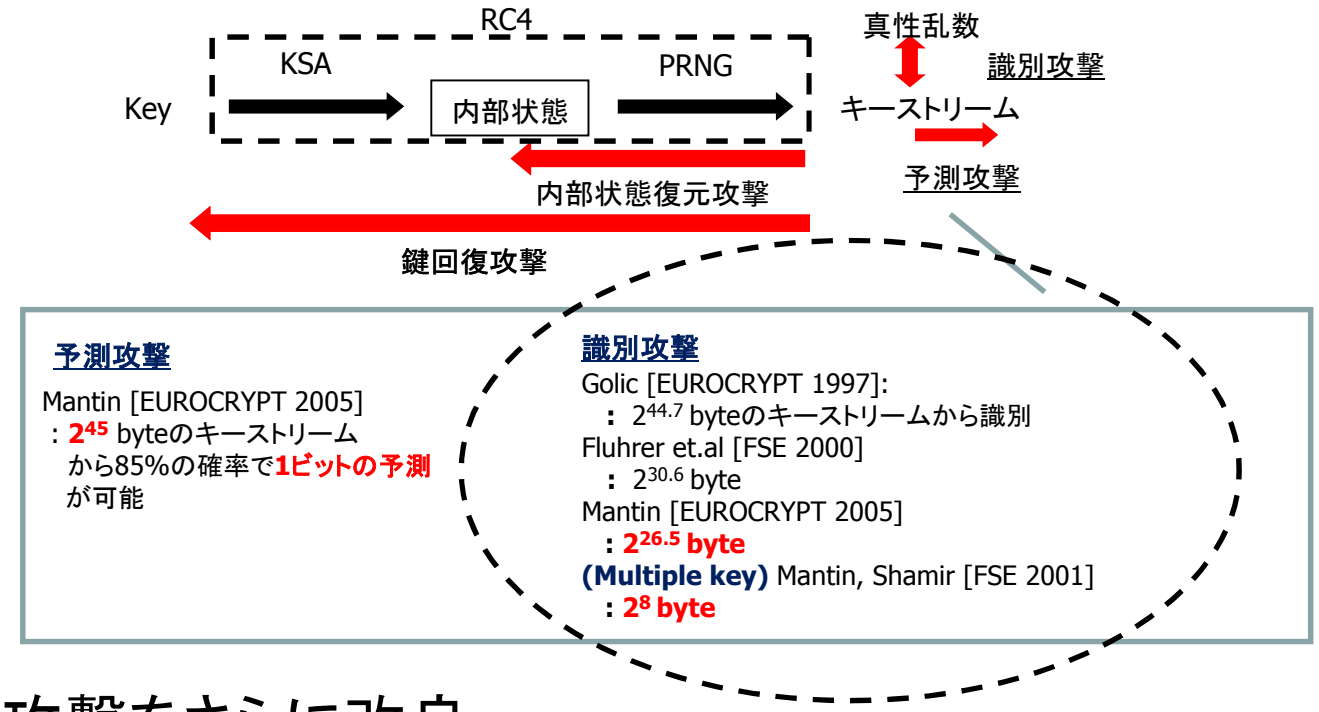
理想的なストリーム暗号としての
安全性を満たしていない

- 実際的な安全性
  - 現実的に脅威となるような攻撃は見つかっていない
    - 簡単に識別できても，鍵回復等の攻撃ができるわけではない
    - 鍵回復攻撃や内部状態推定攻撃の計算量は，非現実的

3. 新しい攻撃法：平文回復攻撃

攻撃のポイント

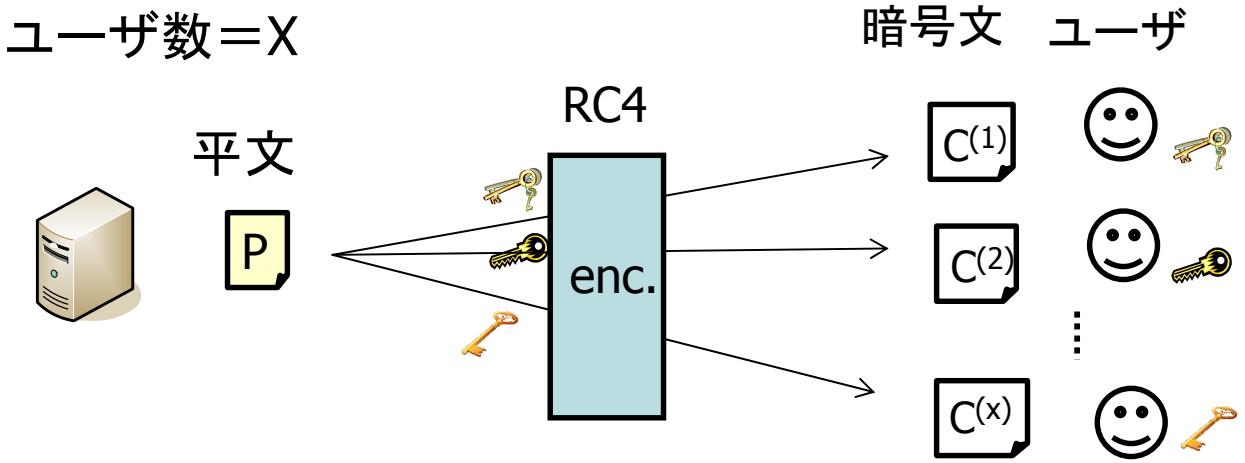
- 現実的なデータ量で実行可能な識別攻撃がベース



- 識別攻撃をさらに改良
  - 新たな出力の偏りを見つけ，理論的にも実験的にも証明
- 効果的な攻撃モデルへの適用
  - Broadcast setting, multi session setting

Broadcast Setting

- 同じ平文 **P** をユーザごとの鍵で暗号化して送信するモデル
  - 攻撃のゴール：攻撃者は暗号文から平文Pを求める



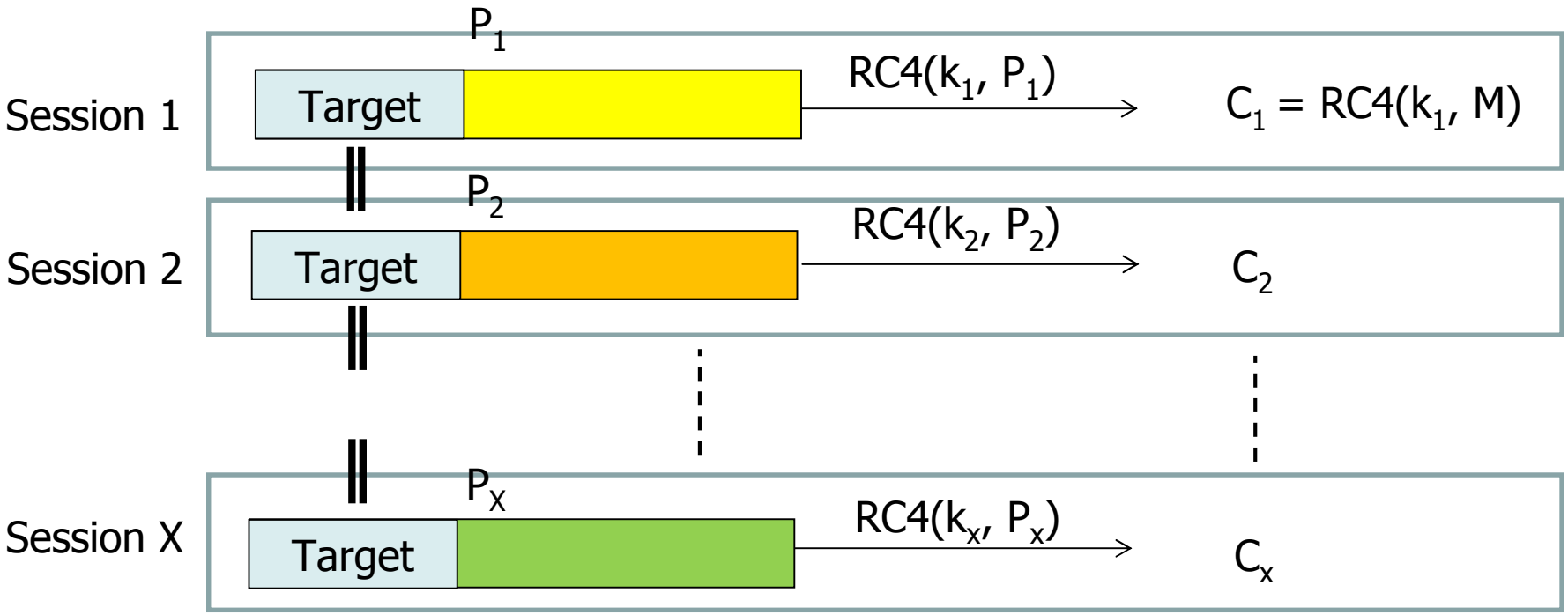
鍵はユーザ毎にランダムに作成

- 例
  1. 複数のユーザが同じデータを取得
  2 同じデータを何度も取得
     => Multi session setting (SSL/TLS)

例）
- **HTTPS + basic認証**
  - ネットワーク利用者認証
  - グループ利用の**Web**ページ
- **OS**イメージの配布など

Multi Session Setting (SSL)

- 異なるsessionにおいて，同じデータを同じpositionで送る場合を想定
  - SSL/TLSでは毎session異なる鍵を生成
  - cookieやpasswordが攻撃Target



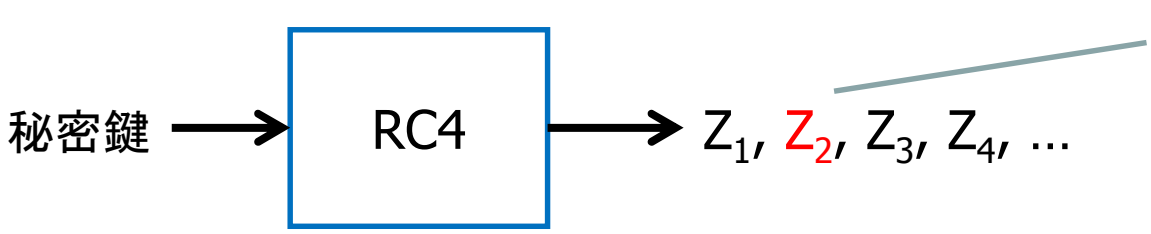
平文回復攻撃 [FSE 2013]

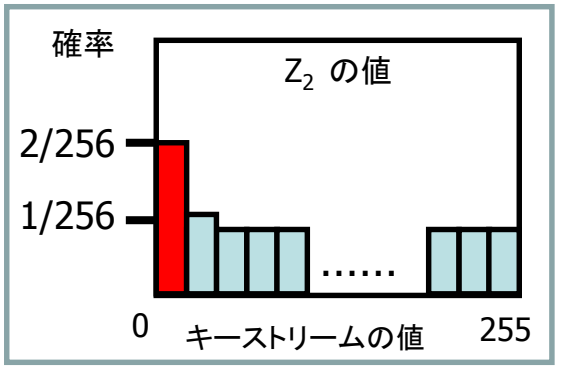
- **攻撃1：平文の初期byte回復攻撃**
  - 2^{32}の暗号文から，
    平文の初期257 byteの任意byte を確率0.5以上で推測可能

- **攻撃2：逐次的平文回復攻撃**
  - 2^{34}の暗号文から，
    平文の連続した初期1000T byteをほぼ確率1で推測可能

攻撃1:平文の初期byte回復攻撃のアイデア

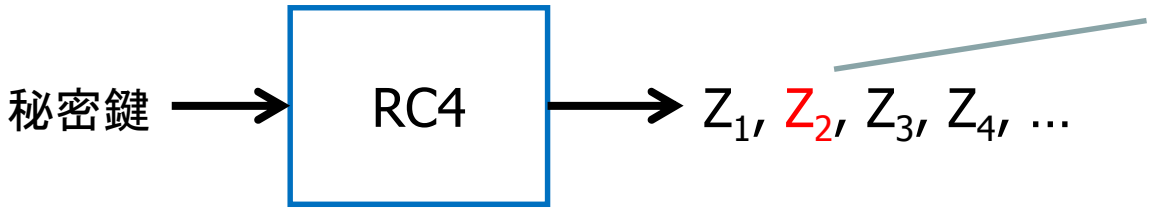
- 出力の偏りから平文回復攻撃へ変換可能 [FSE 2001]
  - キーストリームの2 byte目が0となる確率が2/256



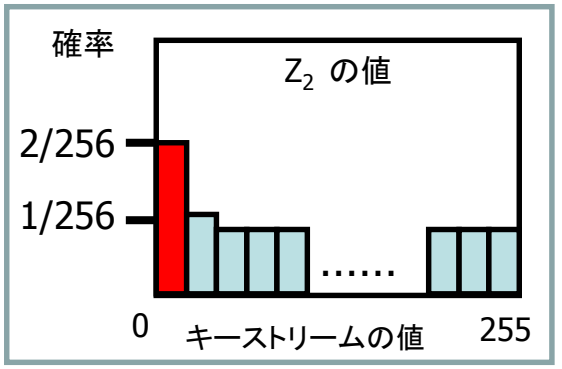


攻撃1:平文の初期byte回復攻撃のアイデア

- 出力の偏りから平文回復攻撃へ変換可能 [FSE 2001]
  - キーストリームの2 byte目が0となる確率が2/256



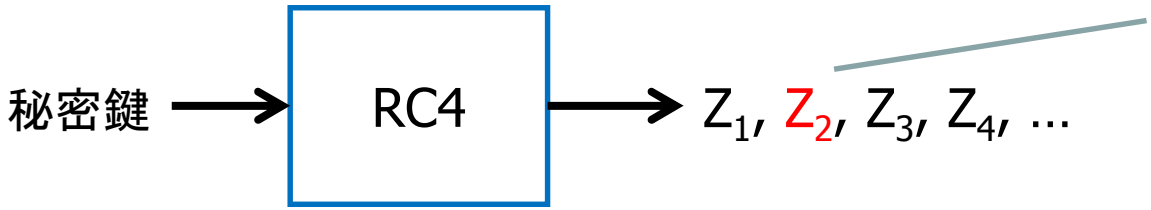
P_r :平文の r byte 目
C_r :暗号文の r byte目



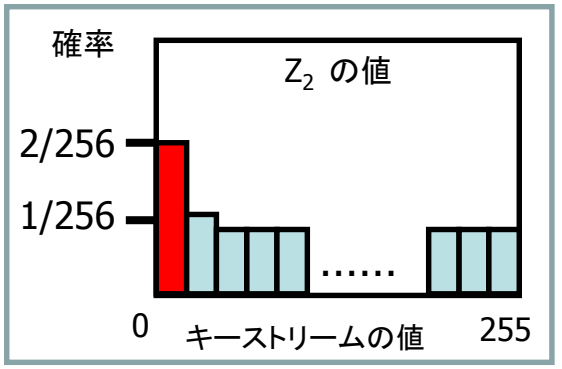
用いる関係式：$C_2 = P_2$ XOR Z_2
→ C_2 =CON XOR Z_2 (Broadcast setting)

攻撃1:平文の初期byte回復攻撃のアイデア

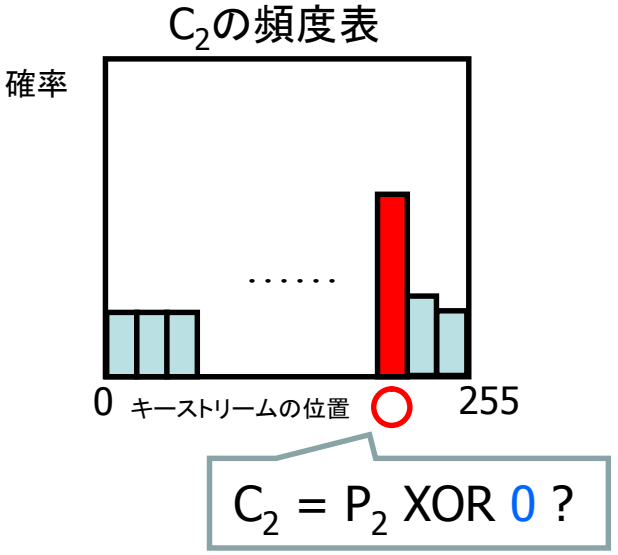
- 出力の偏りから平文回復攻撃へ変換可能 [FSE 2001]
  - キーストリームの2 byte目が0となる確率が2/256



P_r :平文の r byte 目
C_r :暗号文の r byte目

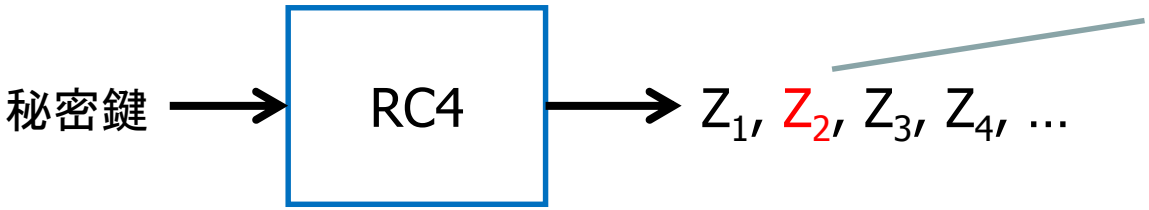


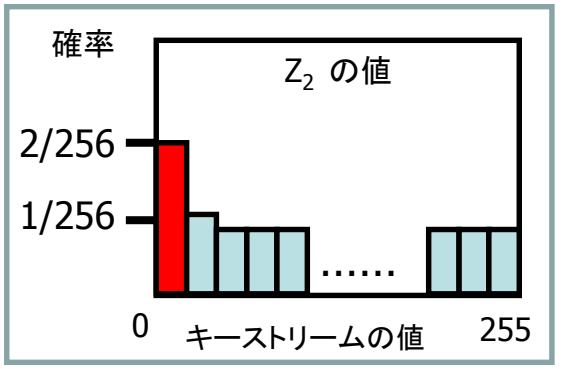
用いる関係式：$C_2 = P_2$ XOR Z_2
→ C_2 =CON XOR Z_2 (Broadcast setting)



攻撃1:平文の初期byte回復攻撃のアイデア

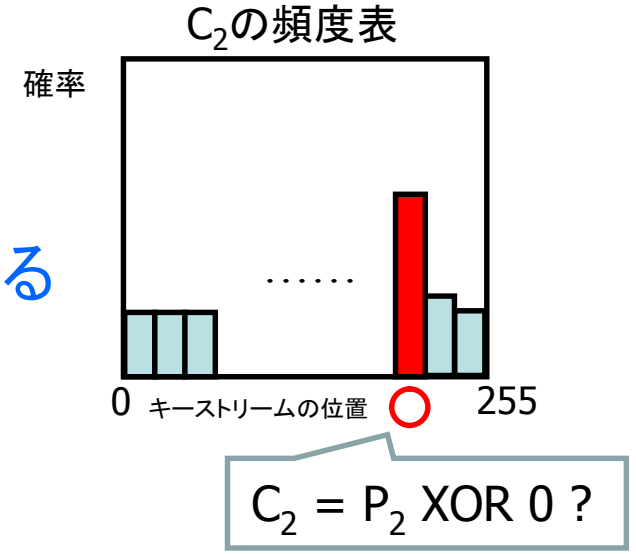
- 出力の偏りから平文回復攻撃へ変換可能 [FSE 2001]
  - キーストリームの2 byte目が0となる確率が2/256





P_r :平文の r byte 目
C_r :暗号文の r byte目

用いる関係式：$C_2 = P_2$ XOR Z_2
→ C_2 =CON XOR Z_2 (Broadcast setting)

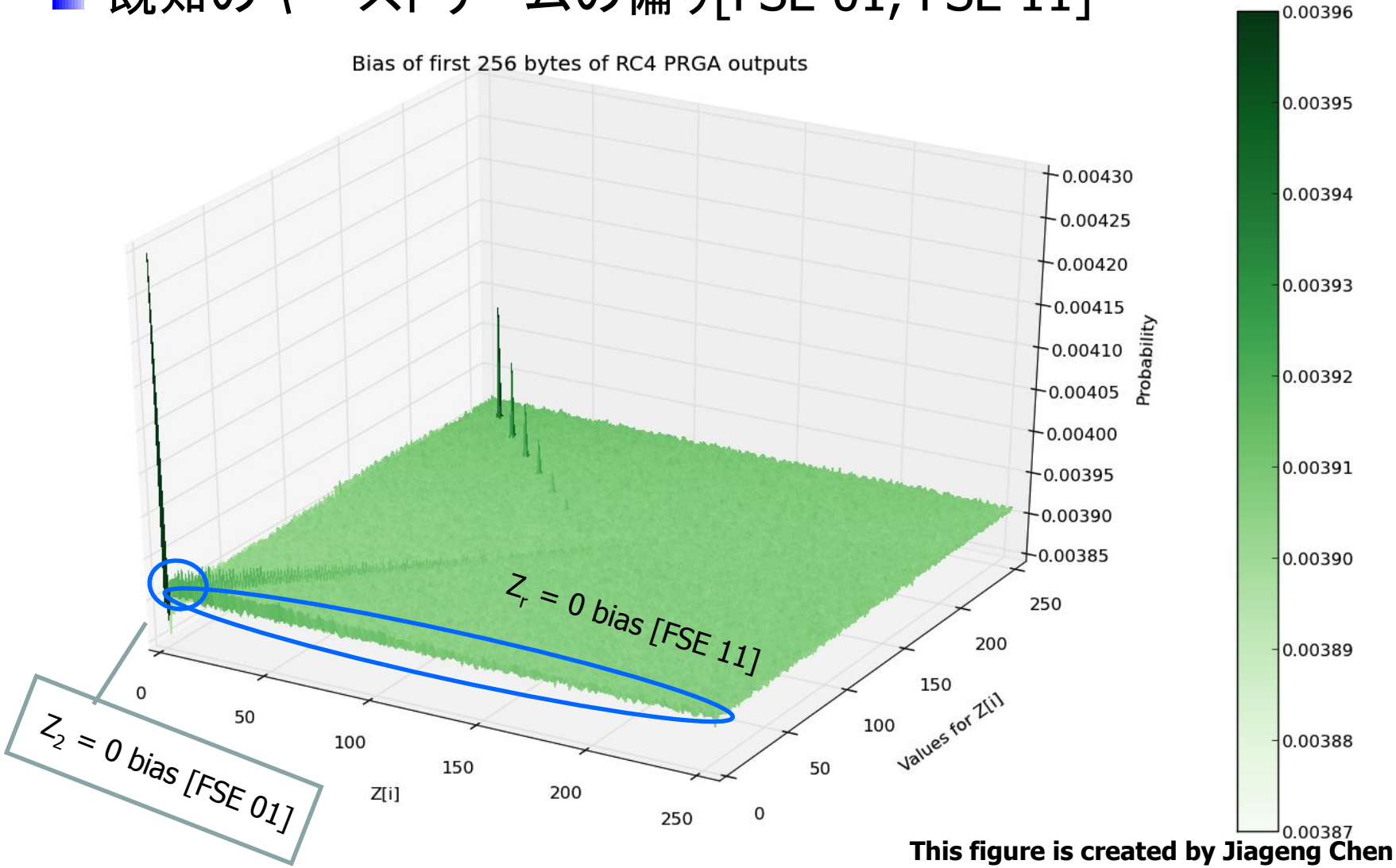


➡ 最も多く出現するC_2がP_2の値と推測される

2^8 通り以上の暗号文があれば十分高い確率で,
暗号文から平文の2 byte目を特定可能

キーストリームの偏り

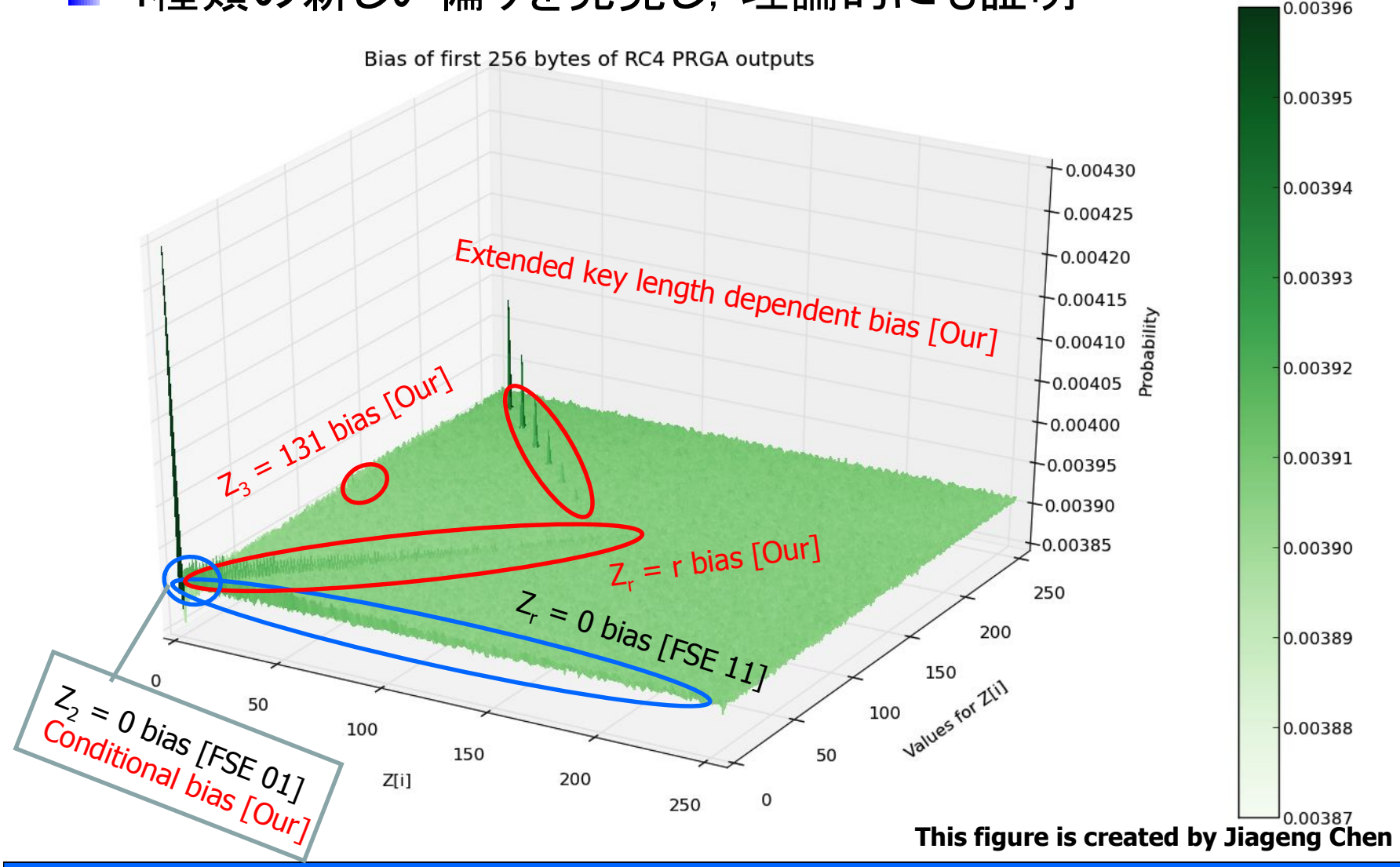
- 既知のキーストリームの偏り[FSE 01, FSE 11]



This figure is created by Jiageng Chen

キーストリームの偏り

- 4種類の新しい偏りを発見し，理論的にも証明



This figure is created by Jiageng Chen

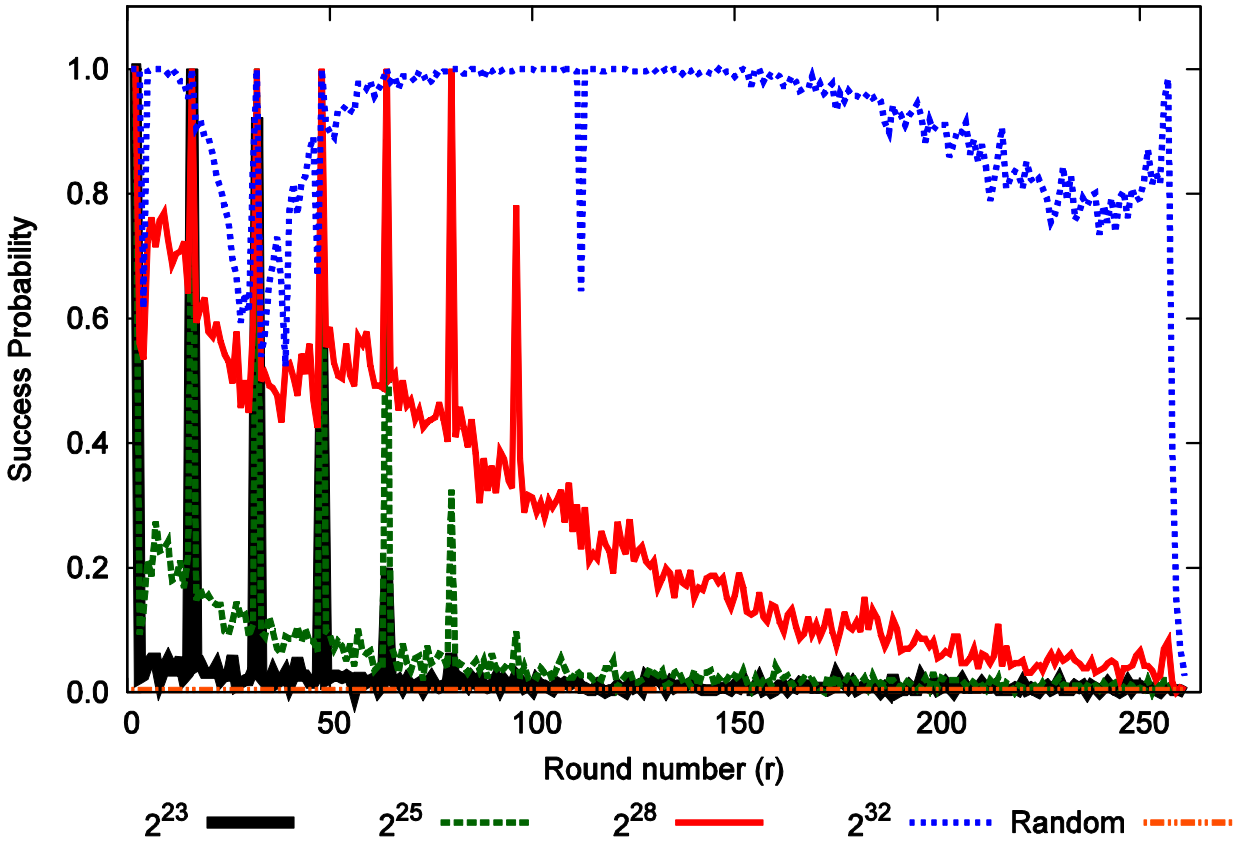
キーストリームの偏り

- 初めの257 byteの最も強い偏り値の実験値と理論値

r	Strongest known bias of Z_r	Prob.(Theoretical)[4]	Prob.(Experimental)
1	$Z_1 = 0 \mid Z_2 = 0$ (Our)	$2^{-8} \cdot (1 + 2^{-1.009})$	$2^{-8} \cdot (1 + 2^{-1.036})$
2	$Z_2 = 0$ [11]	$2^{-8} \cdot (1 + 2^{0})$	$2^{-8} \cdot (1 + 2^{0.002})$
3	$Z_3 = 131$ (Our)	$2^{-8} \cdot (1 + 2^{-8.089})$	$2^{-8} \cdot (1 + 2^{-8.109})$
4	$Z_4 = 0$ [8]	$2^{-8} \cdot (1 + 2^{-7.581})$	$2^{-8} \cdot (1 + 2^{-7.611})$
5–15	$Z_r = r$ (Our)	max: $2^{-8} \cdot (1 + 2^{-7.627})$ min: $2^{-8} \cdot (1 + 2^{-7.737})$	max: $2^{-8} \cdot (1 + 2^{-7.335})$ min: $2^{-8} \cdot (1 + 2^{-7.535})$
16	$Z_{16} = 240$ [5]	$2^{-8} \cdot (1 + 2^{-4.671})$	$2^{-8} \cdot (1 + 2^{-4.811})$
17–31	$Z_r = r$ (Our)	max: $2^{-8} \cdot (1 + 2^{-7.759})$ min: $2^{-8} \cdot (1 + 2^{-7.912})$	max: $2^{-8} \cdot (1 + 2^{-7.576})$ min: $2^{-8} \cdot (1 + 2^{-7.839})$
32	$Z_{32} = 224$ (Our)	$2^{-8} \cdot (1 + 2^{-5.176})$	$2^{-8} \cdot (1 + 2^{-5.383})$
33–47	$Z_r = 0$ [8]	max: $2^{-8} \cdot (1 + 2^{-7.897})$ min: $2^{-8} \cdot (1 + 2^{-8.050})$	max: $2^{-8} \cdot (1 + 2^{-7.868})$ min: $2^{-8} \cdot (1 + 2^{-8.039})$
48	$Z_{48} = 208$ (Our)	$2^{-8} \cdot (1 + 2^{-5.651})$	$2^{-8} \cdot (1 + 2^{-5.938})$
49–63	$Z_r = 0$ [8]	max: $2^{-8} \cdot (1 + 2^{-8.072})$ min: $2^{-8} \cdot (1 + 2^{-8.224})$	max: $2^{-8} \cdot (1 + 2^{-8.046})$ min: $2^{-8} \cdot (1 + 2^{-8.238})$
64	$Z_{64} = 192$ (Our)	$2^{-8} \cdot (1 + 2^{-6.085})$	$2^{-8} \cdot (1 + 2^{-6.496})$
65–79	$Z_r = 0$ [8]	max: $2^{-8} \cdot (1 + 2^{-8.246})$ min: $2^{-8} \cdot (1 + 2^{-8.398})$	max: $2^{-8} \cdot (1 + 2^{-8.223})$ min: $2^{-8} \cdot (1 + 2^{-8.376})$
80	$Z_{80} = 176$ (Our)	$2^{-8} \cdot (1 + 2^{-6.574})$	$2^{-8} \cdot (1 + 2^{-7.224})$
81–95	$Z_r = 0$ [8]	max: $2^{-8} \cdot (1 + 2^{-8.420})$ min: $2^{-8} \cdot (1 + 2^{-8.571})$	max: $2^{-8} \cdot (1 + 2^{-8.398})$ min: $2^{-8} \cdot (1 + 2^{-8.565})$
96	$Z_{96} = 160$ (Our)	$2^{-8} \cdot (1 + 2^{-6.970})$	$2^{-8} \cdot (1 + 2^{-7.911})$
97–111	$Z_r = 0$ [8]	max: $2^{-8} \cdot (1 + 2^{-8.592})$ min: $2^{-8} \cdot (1 + 2^{-8.741})$	max: $2^{-8} \cdot (1 + 2^{-8.570})$ min: $2^{-8} \cdot (1 + 2^{-8.722})$
112	$Z_{112} = 144$ (Our)	$2^{-8} \cdot (1 + 2^{-7.300})$	$2^{-8} \cdot (1 + 2^{-8.666})$
113–255	$Z_r = 0$ [8]	max: $2^{-8} \cdot (1 + 2^{-8.763})$ min: $2^{-8} \cdot (1 + 2^{-10.052})$	max: $2^{-8} \cdot (1 + 2^{-8.760})$ min: $2^{-8} \cdot (1 + 2^{-10.041})$
256	$Z_r = 0$(*negative bias*) (Our)	N/A	$2^{-8} \cdot (1 - 2^{-9.407})$
257	$Z_r = 0$ (Our)	N/A	$2^{-8} \cdot (1 + 2^{-9.531})$

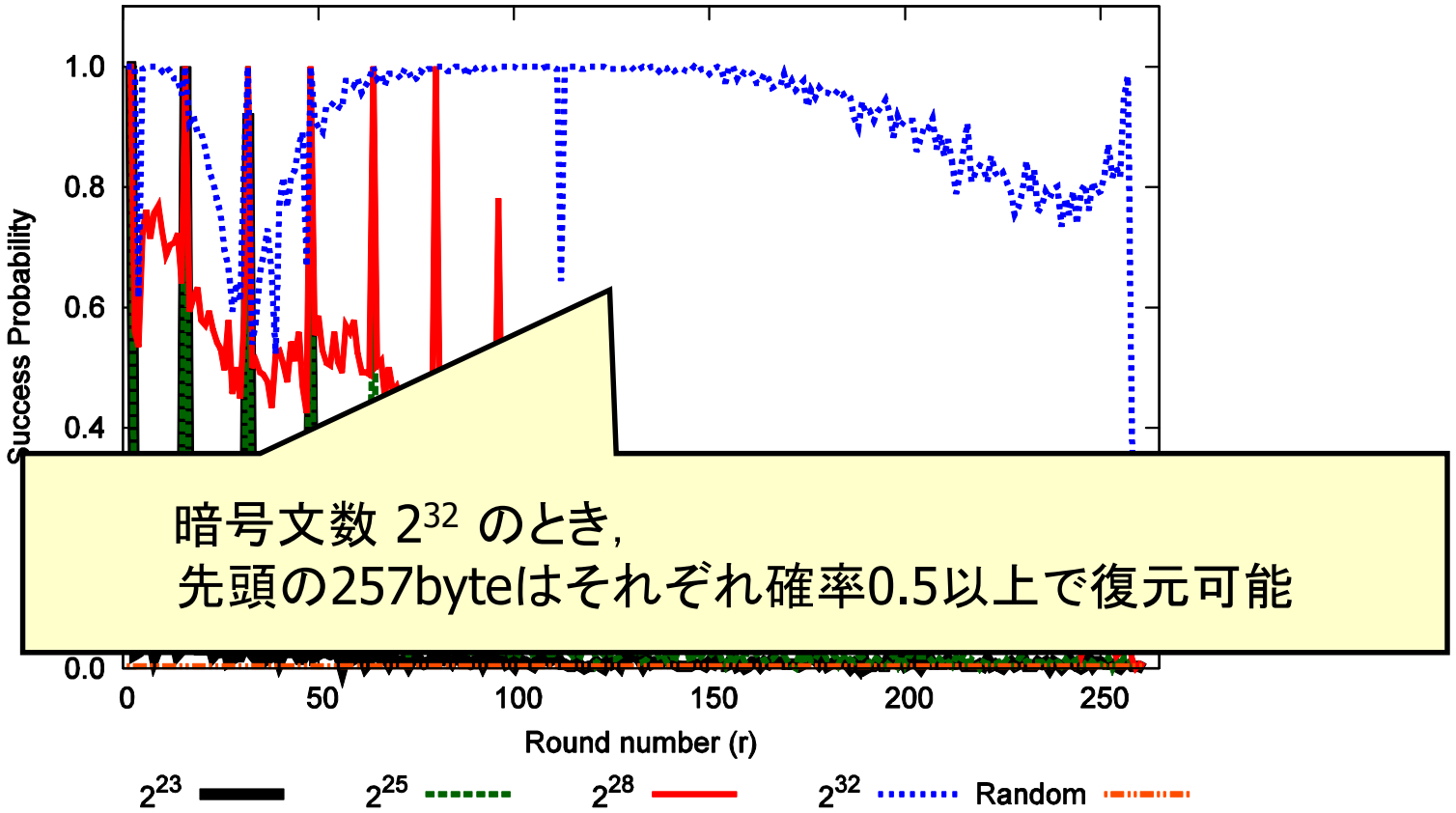
攻撃1の実験結果

- 各byteにおける平文復元の成功確率
  - 暗号文数：2^{24}, 2^{28}, 2^{32}, 2^{35}
  - ランダムに生成した256通りの平文に対して実施



攻撃1の実験結果

- 各byteにおける平文復元の成功確率
  - 暗号文数：2^{24}, 2^{28}, 2^{32}, 2^{35}
  - ランダムに生成した256通りの平文に対して実施



暗号文数 2^{32} のとき，
先頭の257byteはそれぞれ確率0.5以上で復元可能

平文回復攻撃 [FSE 2013]

- **攻撃1：平文の初期byte回復攻撃**
  - 2^{32}の暗号文から，
    平文の初期257 byteの任意byte を確率0.5以上で推測可能

- **攻撃2：逐次的平文回復攻撃**
  - 2^{34}の暗号文から，
    平文の連続した初期1000T byteをほぼ確率1で推測可能

攻撃2：逐次的平文回復攻撃

- 258バイト目以降の平文を求める方法

P_1 P_2 P_3 … P_{31} … P_{257} | $P_{258}, P_{259}, \cdots$

ここまでは初期の特有の偏りを利用　　初期のような強い偏りが存在しない

- 任意のbyteで発生する**long term bias**を利用
  - Digraph Repetition Bias (call ABSAB bias) [EUROCRYPT 2005]
    - 既知の最も強力なlong-term bias
    - Gバイトのギャップの後に同じパターンが生じる（2バイト単位）

キーストリーム　….ABHLWECTSDGAB….

gap G

攻撃2の評価

- 攻撃方法
  - 257 byteを求めたあと，ABSAB biasにより，逐次的に求めていく

P_1 P_2 P_3 … P_{31} … P_{257} P_{258}, P_{259},…

2^{32}の暗号文で高確率で復元可能(攻撃1)

ABSAB bias

- 計算機実験
  - 4バイト(P_{258}, …, P_{261})を逐次的に復元したときの成功確率

Table 1: Success Probability of our algorithm for recovering P_r ($r \geq 258$) on Broadcast RC4

	# of ciphertexts				
	2^{30}	2^{31}	2^{32}	2^{33}	2^{34}
P_{258}	0.0039	0.0391	0.3867	0.9648	1.0000
P_{259}	0.0039	0.0078	0.1523	0.9414	1.0000
P_{260}	0.0000	0.0039	0.0703	0.9219	1.0000
P_{261}	0.0000	0.0078	0.0273	0.9023	1.0000

- 理論値
  - 暗号文数 2^{34} のとき、平文の先頭 2^{50} ≒ 1000 T bytesを確率0.97で復元可能(識別攻撃 Pr = $1 - 2^{-19}$, Xバイトの復元の成功確率 $(1 - 2^{-19})^{255 \cdot X}$)

新しい攻撃法(平文回復攻撃)まとめ

- 攻撃の条件
  - 平文が異なる鍵で暗号化(Broadcast setting).
    - SSL/TLSでは，毎session同じ位置(multi session setting)
  - 攻撃者は暗号文を集めるのみ (暗号文単独攻撃)
- 攻撃能力
  - 2^{24}-2^{35}の暗号文から，平文を高確率で求めることができる.
- 実際の影響
  - 2^{24}-2^{32}の暗号文が必要であるため，すぐさま脅威になることはない.
  - ほかの脆弱性と組み合わさってPracticalになる可能性あり.
    - HTTPSリクエストを大量にするJavascript等の利用

4. その後の進展

RC4の攻撃の進展

FSE 2013での発表以降さまざまな攻撃の改良が行われた

- SSL/TLSへの平文回復攻撃の改良
  - 現実的な平文パターンでの評価 [ICSS 2013]
  - 比較的安全な実装方法(RC4-drop)への拡張 [SAC 2013]
  - 攻撃の確率向上 [USENIX 2013]
- WPA-TKIPへの攻撃の拡張
  - 平文回復攻撃 [FSE 2014]

平文空間を制限した場合における平文回復攻撃 [ICSS 2013]

- FSE 2013では各バイトにランダムな値（256通りの値）が
  が代入されるとして攻撃
  → 実際はある特定の平文空間で使用（パスワード等）

- 平文の候補の条件
  - Case 1 : PIN code (0 – 9, 0x30 – 0x39, 10種類)
  - Case 2 : ASCII code (except control code, 0x20 – 0x7e, 95種類)
  - Case 3 : Randomly distributed (256種類)

ASCII code

Case 1 (PIN code)

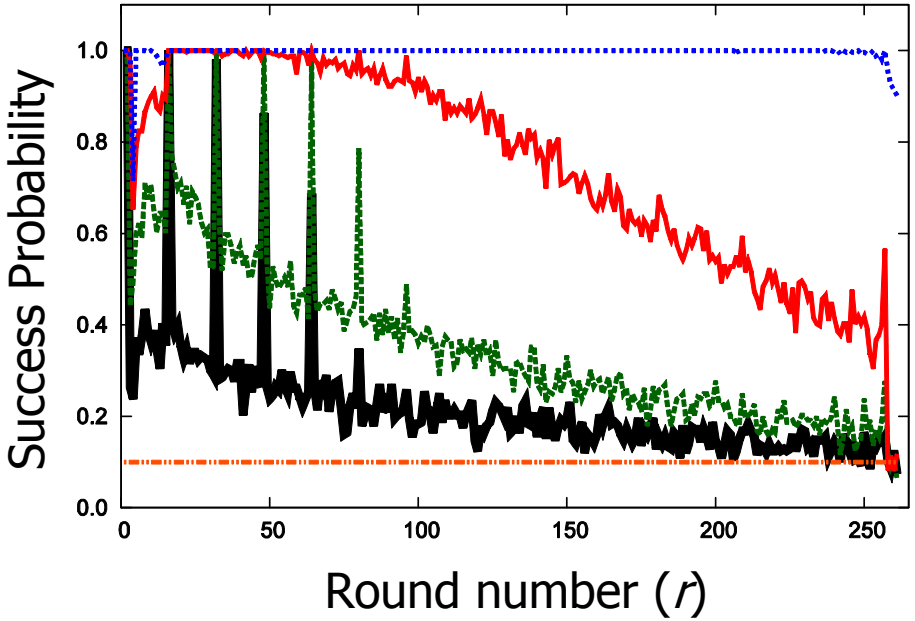
Case 2 (ASCII code except control code)

文字	10進	16進	文字	10進	16進	文字	10進	16進	文字	10進	16進	文字	10進	16進	文字	10進	16進	文字	10進	16進
NUL	0	00	DLE	16	10	SP	32	20	0	48	30	@	64	40	P	80	50	`	96	60	p	112	70
SOH	1	01	DC1	17	11	!	33	21	1	49	31	A	65	41	Q	81	51	a	97	61	q	113	71
STX	2	02	DC2	18	12	"	34	22	2	50	32	B	66	42	R	82	52	b	98	62	r	114	72
ETX	3	03	DC3	19	13	#	35	23	3	51	33	C	67	43	S	83	53	c	99	63	s	115	73
EOT	4	04	DC4	20	14	$	36	24	4	52	34	D	68	44	T	84	54	d	100	64	t	116	74
ENQ	5	05	NAK	21	15	%	37	25	5	53	35	E	69	45	U	85	55	e	101	65	u	117	75
ACK	6	06	SYN	22	16	&	38	26	6	54	36	F	70	46	V	86	56	f	102	66	v	118	76
BEL	7	07	ETB	23	17	'	39	27	7	55	37	G	71	47	W	87	57	g	103	67	w	119	77
BS	8	08	CAN	24	18	(	40	28	8	56	38	H	72	48	X	88	58	h	104	68	x	120	78
HT	9	09	EM	25	19	)	41	29	9	57	39	I	73	49	Y	89	59	i	105	69	y	121	79
LF*	10	0a	SUB	26	1a	*	42	2a	:	58	3a	J	74	4a	Z	90	5a	j	106	6a	z	122	7a
VT	11	0b	ESC	27	1b	+	43	2b	;	59	3b	K	75	4b	[	91	5b	k	107	6b	{	123	7b
FF*	12	0c	FS	28	1c	,	44	2c	<	60	3c	L	76	4c	\¥	92	5c	l	108	6c	|	124	7c
CR	13	0d	GS	29	1d	-	45	2d	=	61	3d	M	77	4d	]	93	5d	m	109	6d	}	125	7d
SO	14	0e	RS	30	1e	.	46	2e	>	62	3e	N	78	4e	^	94	5e	n	110	6e	~	126	7e
SI	15	0f	US	31	1f	/	47	2f	?	63	3f	O	79	4f	_	95	5f	o	111	6f	DEL	127	7f

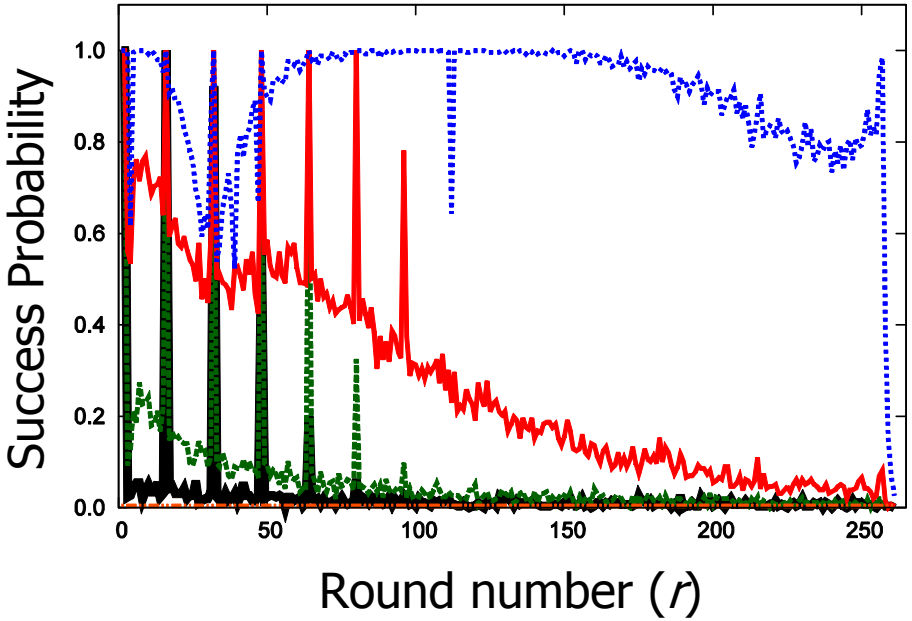
ASCII文字コード, available at : http://e-words.jp/p/r-ascii.html

実験結果-Case 1 & Case 3

・Case 1：PIN code

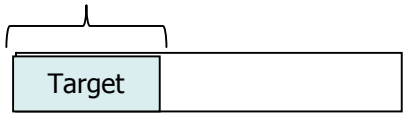


・Case 3：Randomly distributed



暗号文数　2^{23}　2^{25}　2^{28}　2^{32}　Random

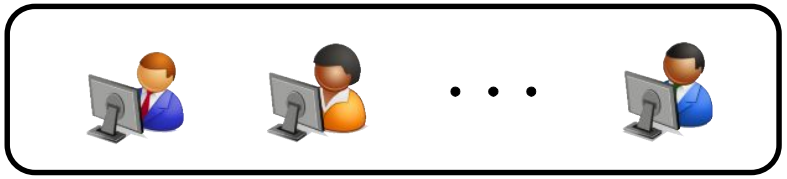
初めの257 bytes



平文

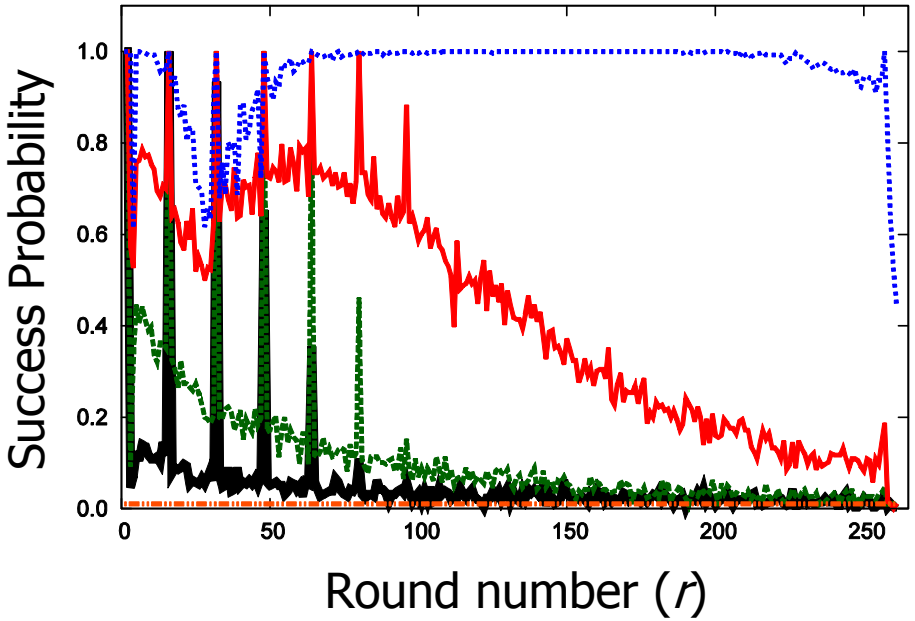
Random guessより高い確率

2^{23} sessions（暗号文）

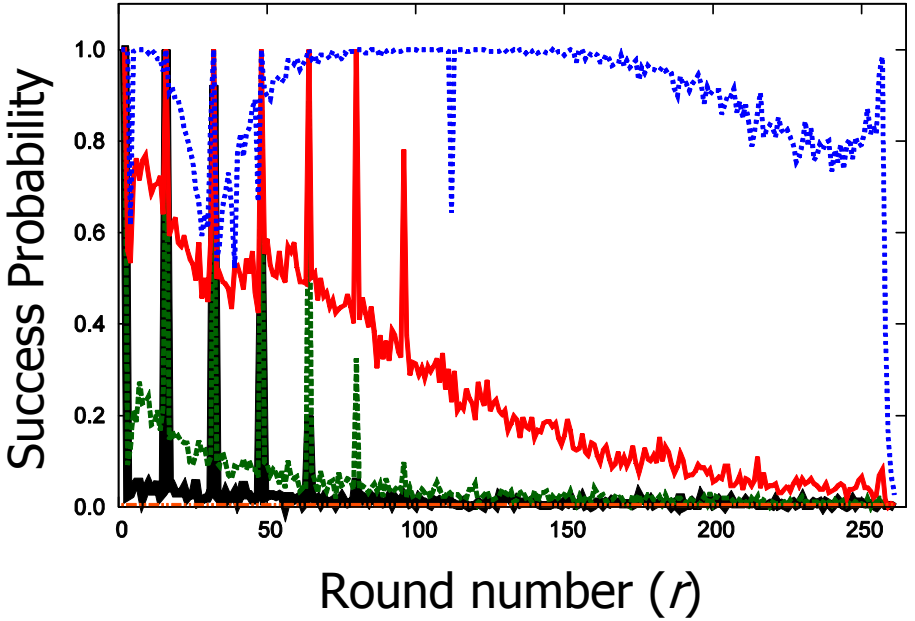


実験結果-Case 2 & Case 3

・Case 2：ASCII code

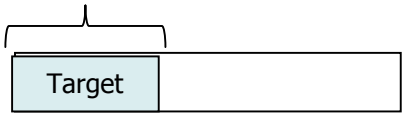


・Case 3：Randomly distributed



暗号文数　2^{23}　2^{25}　2^{28}　2^{32}　Random

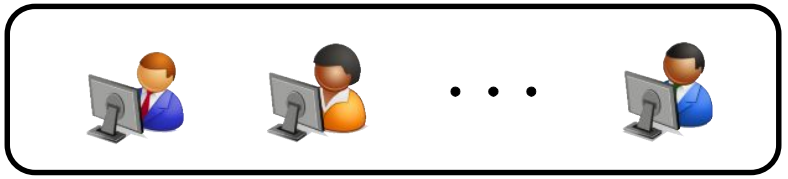
初めの257 bytes



平文

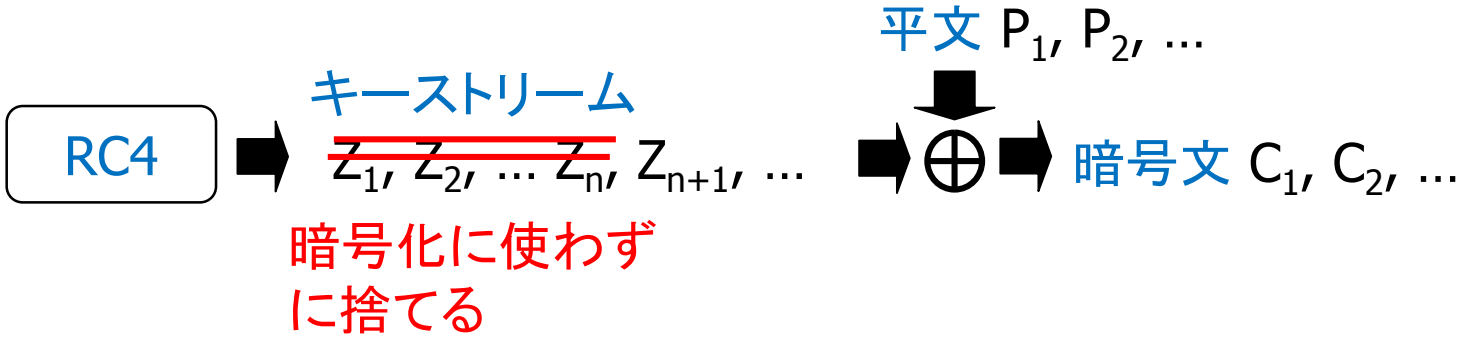
Random guessより高い確率

2^{25} sessions (暗号文)



比較的安全な実装方法(RC4-drop)への拡張 [SAC 2013]

- FSE 2013の攻撃 に強い実装 ”RC4-drop(n)” への攻撃
  - RC4-drop(n): キーストリームの先頭の n バイトを捨てる
    （推奨パラメータ n =768, 理想的には n =3072以上 [CRYPTO 2002]）
    →初期のbiasは排除される



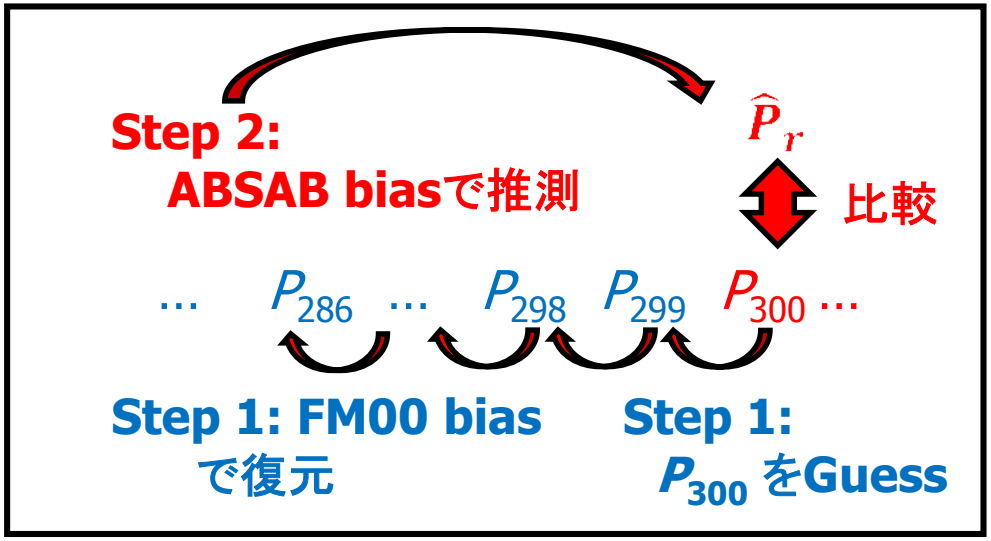
RC4のキーストリームの初期のbiasが取り除かれる

FSE2013の攻撃を含む従来の攻撃が無効

比較的安全な実装方法(RC4-drop)への拡張 [SAC 2013]

- 任意のbyteに存在する複数のbiasを組み合わせて利用
  - Mantin's bias [EURO05] とFluhrer-McGrew bias [FSE00]
  - Initial keystreamを排除してもworkする攻撃

Guess and determine technique



	攻撃に利用できる暗号文数	
	2^{34}	2^{35}
P_{300}	0.8867	1.0000

攻撃成功確率の改良 [USENIX 2013]

- 基本的には，FSE 2013の攻撃手法と同様
  - 初期キーストリームの偏りから，平文回復攻撃
  - 実験結果のみで，偏りの理論的考察はない
- 改善ポイント
  - FSE 2013：もっとも強い偏りの値を推測に利用
  - USENIX 2013：各byteの偏っている分布すべてを利用
- 結果
  - 先頭256バイトの平文を2^{32}個の暗号文から確率0.96以上で回復できる（FSE 2013は0.5以上）
  - 任意バイトを2^{34}の暗号文から確率0.99程度で回復できる（SAC 2013は0.89程度）

WPA-TKIPへの拡張 [FSE 2014]

- WPA-TKIP
  - 無線LANの暗号化方式でRC4を利用
  - WEP鍵更新方法を変更：TKIP (Temporal Key Integrity Protocol)
  - 新しい偏りが出現 => 3 byteのIVの影響

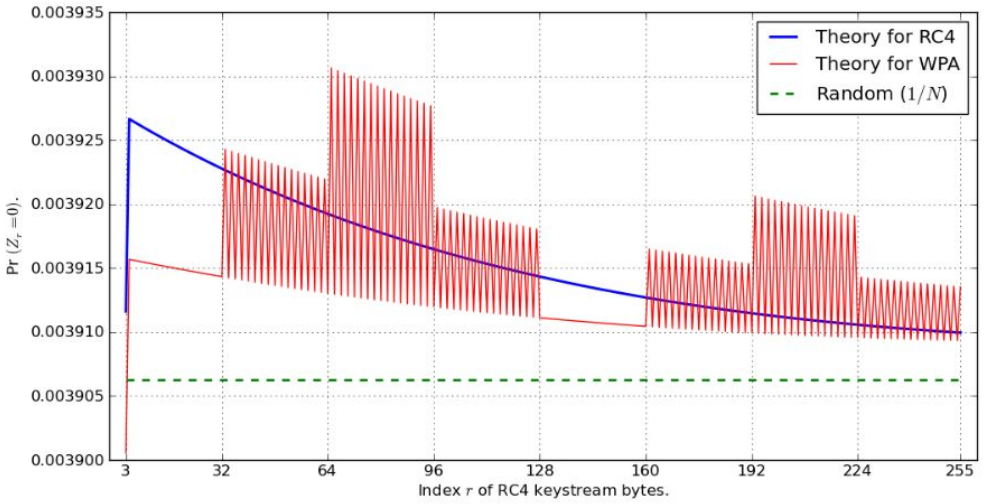


Fig. 5. Theoretical plot for $\Pr(Z_r - 0)$ for RC4 and WPA, where $r - 3, \ldots, 255$.

[ePrint 2013]

- 攻撃
  - 初めの256 byteは, SSL/TLSと同様に2^{30}程度で回復可能

まとめ

- RC4の安全性
  - ストリーム暗号としての安全性は満たしていない
    - Practicalな識別攻撃, weak keyの存在
- SSL/TLS–RC4 への攻撃
  - 2^{24} - 2^{35}の程度の暗号文から平文は特定可能
    - 平文空間を限定するとさらに効率化可能
- 対策
  - 攻撃者は，暗号文を集めるのみでいいので，基本的にRC4を使わない以外の対策はない
  - SSL/TLSでは，CBC modeもBEAST, Lucky Thirteen, CRIME等の脆弱性が報告されているため，

    => Authenticated Encryptionの利用

References

[FSE 2013] T. Isobe, T. Ohigashi, Y. Watanabe and M. Morii, "Full Plaintext Recovery Attack on Broadcast RC4"

[ICSS 2013] Y. Watanabe, T. Isobe, T. Ohigashi, M. Morii, "Vulnerability of RC4 in SSL/TLS"

[SAC 2013] T. Ohigashi, T. Isobe, Y. Watanabe and M. Morii, "How to Recover Any Byte of Plaintexton RC4"

[USENIX 2013] N. J. AlFardan, D. J. Bernstein, K. G. Paterson, B. Poettering and J. C. N. Schuldt, "On the Security of RC4 in TLS"

[SAC 2013] T. Ohigashi, T. Isobe, Y. Watanabe and M. Morii, "How to Recover Any Byte of Plaintexton RC4"

[FSE 2014] K. G. Paterson, J. C. N. Schuldt and B. Poettering, "Plaintext Recovery Attacks Against WPA/TKIP"

[EUROCRYPT 2005] I. Mantin, "Predicting and Distinguishing Attacks on RC4 Keystream Generator"

[EUROCRYPT 1997] J. D. Golic, "Linear Statistical Weakness of Alleged RC4 Key-Stream Generator"

[FSE 2000] S. R. Fluhrer and D. A. McGrew, "Statistical Analysis of the Alleged RC4 Keystream Generator"

[FSE 2001] I. Mantin and A. Shamir, "A Practical Attack on Broadcast RC4"

[ASIACRYPT 1998] L. R. Knudsen, W. Meier, B. Preneel, V. Rijmen, S. Verdoolaege, "Analysis methods for (alleged) RC4"

[IEICE 2003] Y. Shiraishi, T. Ohigashi, and M.Morii, "Internal-State Reconstruction of a Stream Cipher RC4"

[CRYPTO 2008] A.Maximov and D. Khovratovich, "New State Recovery Attack on RC4"

[R'1995] A. Roos, "Class of weak keys in the RC4 stream cipher" Two posts in sci.crypt, 1995

[SAC 2010] P.Sepehrdad, S. Vaudenay, and M. Vuagnoux, "Discovery and Exploitation of New Biases in RC4 Discovery and Exploitation ofNew Biases in RC4"

[JIP 2014] A. Nagao, T. Ohigashi, T. Isobe, and M. Morii, "Expanding Weak-Key Space of RC4,"

[FSE 2011] S. Maitra, G. Paul, and S. Sen Gupta, "Attack on Broadcast RC4 revisit"

[CRYPTO 2002] I .Mironov, "(Not so) Random Shuffles of RC4"

[ePrint 2013] S. Sen Gupta, S. Maitra, W. Meier, G.Paul and S. Sarkar "Some results on RC4 in WPA"

BEASTとの比較

- 仮定1 (HTTPリクエスト大量に生成可能)
  - RC4 ：最悪2^{34} 程度で攻撃可能
  - BEAST：攻撃不可
- 仮定2 (HTTPリクエスト大量に生成可能 + リクエストPOSTの長さをコントロール可能)
  - RC4 ：場所の最適化で2^{34} 以下の攻撃は可能
  - BEAST：攻撃不可
- 仮定3 (HTTPリクエスト大量に生成可能 + リクエストPOSTの長さをコントロール可能　+　リクエストの一部を改ざん可能)
  - RC4 ：場所の最適化で2^{34} 以下の攻撃は可能
  - BEAST：最悪2^{8} 程度で攻撃可能 (byte 単位guess)