28 August, 2013
ASK 2013 @ Weihai, China

Recent Cryptanalysis of RC4 Stream Cipher

Takanori Isobe
Kobe University

Joint work with
Toshihiro Ohigashi, Yuhei Watanabe,
and Maskatu Morii

Agenda

This talk contains two results

1. Initial Keystream Biases of RC4 and Its Applications
   (From FSE 2013)
   #The full version will appear in IEICE journal 2014

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

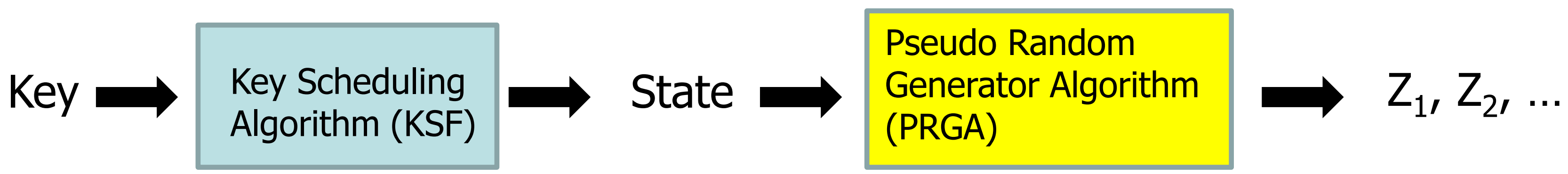
-T. Isobe, T. Ohigashi, Y. Watanabe, M. Morii "Comprehensive Analysis of Initial Keystream Biases of RC4", IEICE Journal, to appear

2. Advanced Plaintext Recovery Attacks on RC4
   (From SAC 2013)

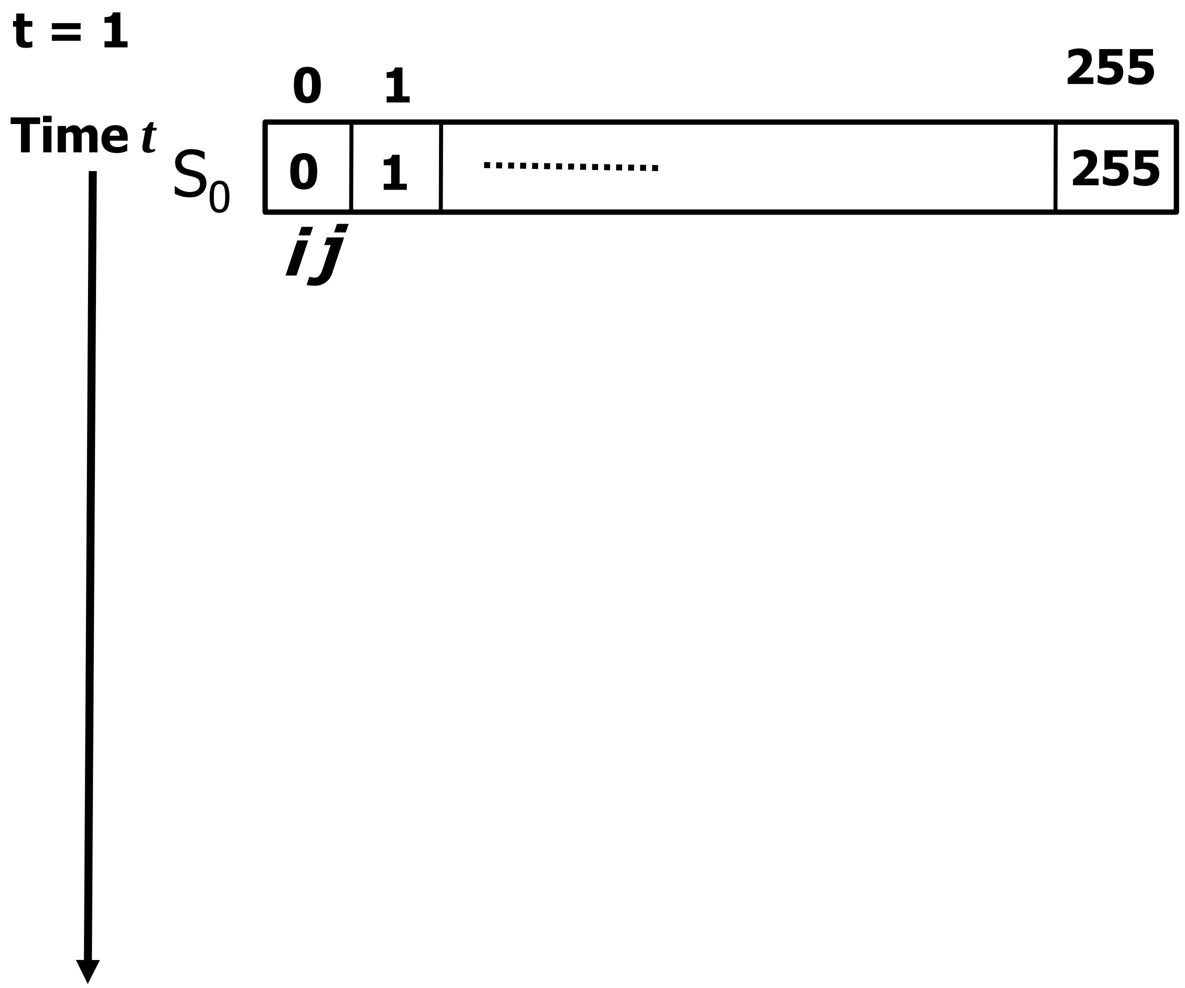
-T. Ohigashi, T. Isobe, Y. Watanabe, M. Morii "How to Recover Any Bytes of Plaintext on RC4", SAC 2013

RC4 Stream Cipher

- Stream cipher designed by Rivest in 1987
  - One of most famous stream ciphers
    - SSL/TLS, WEP/WPA and more
- Typical Parameter
  - Key size : 16 bytes (128 bits)
  - State size : 256 bytes
- Consist of
  - Key Scheduling Algorithm (KSA)
  - Pseudo Random Generator Algorithm (PRGA)

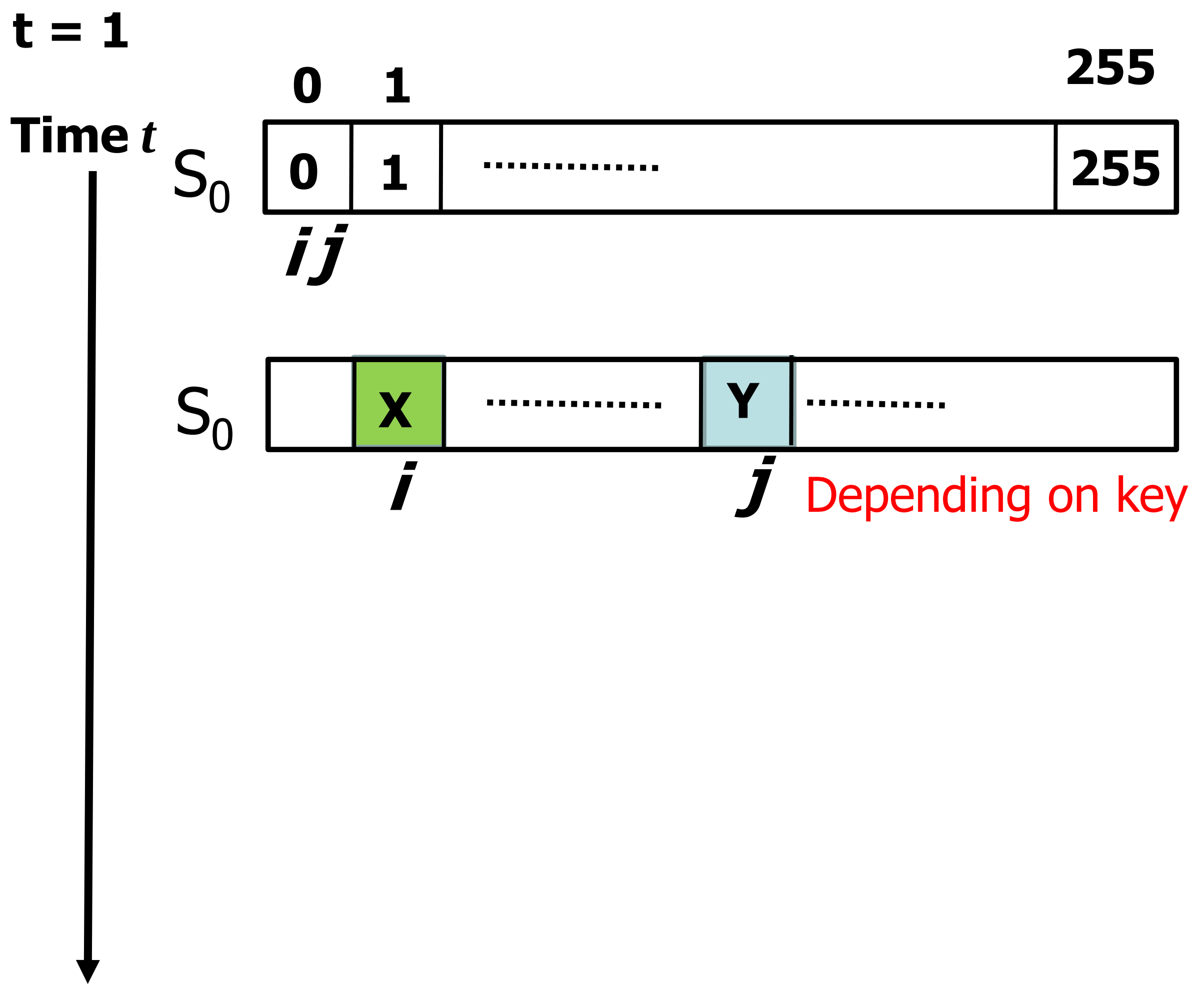


Key Scheduling Algorithm



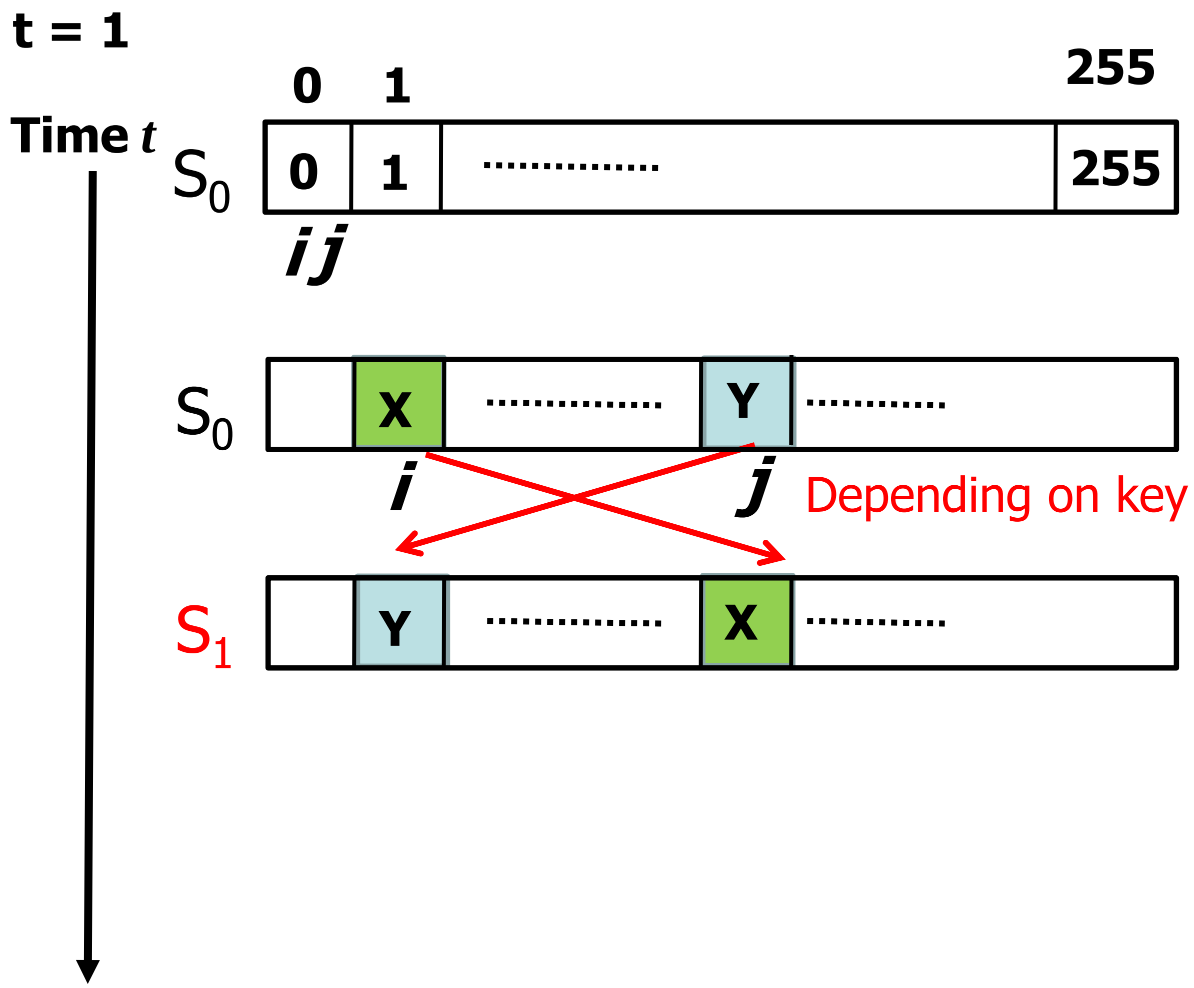

```
i = 0
J = 0
S0[x] = x
loop
  j = j +S[ i ] + K[i]
  swap(S[ i ], S[ j ])
  i = I + 1
end loop
```

Key Scheduling Algorithm



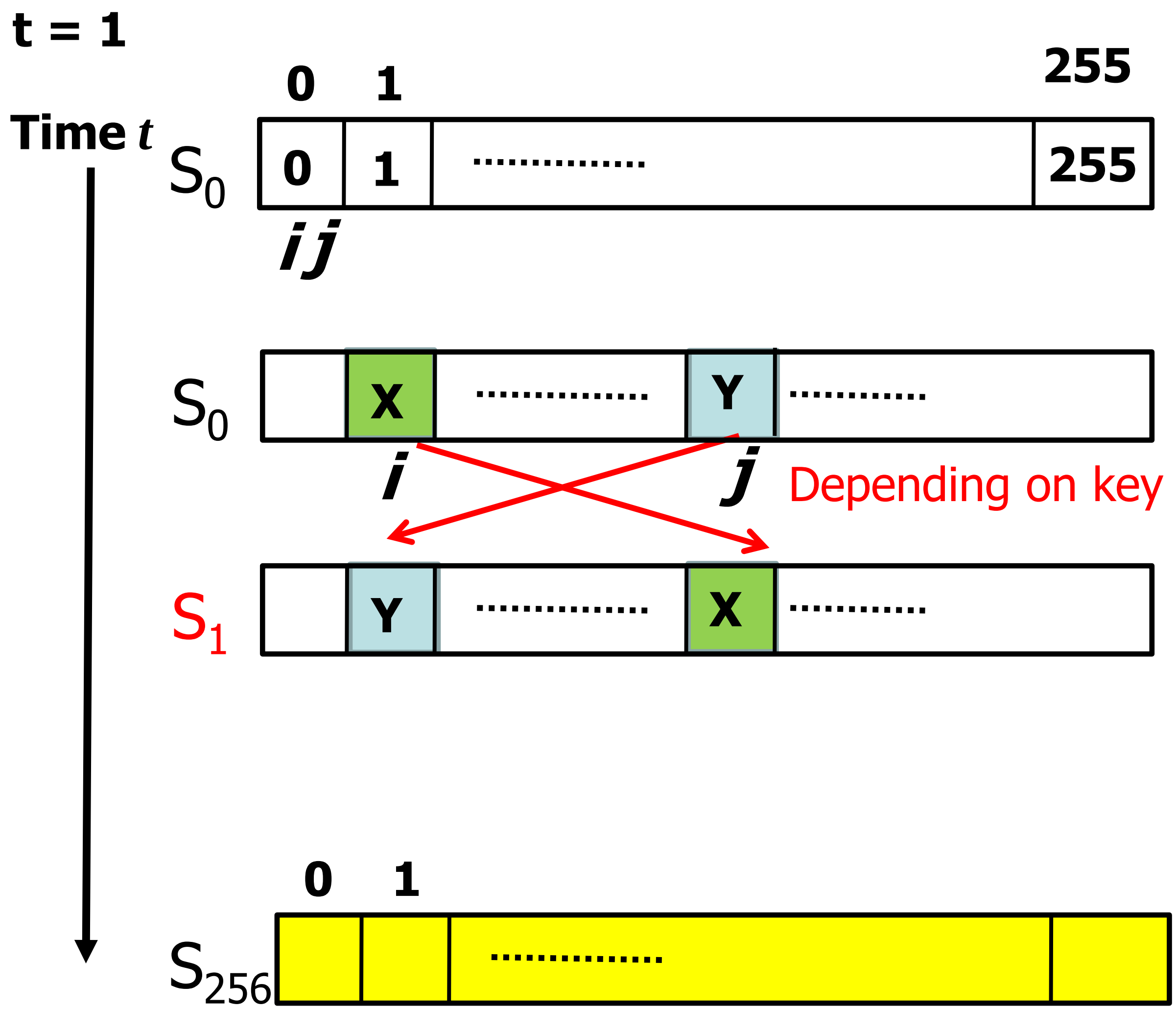

```
i = 0
J = 0
S0[x] = x
loop
  j = j +S[ i ] + K[i]
  swap(S[ i ], S[ j ])
  i = I + 1
end loop
```

Key Scheduling Algorithm




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

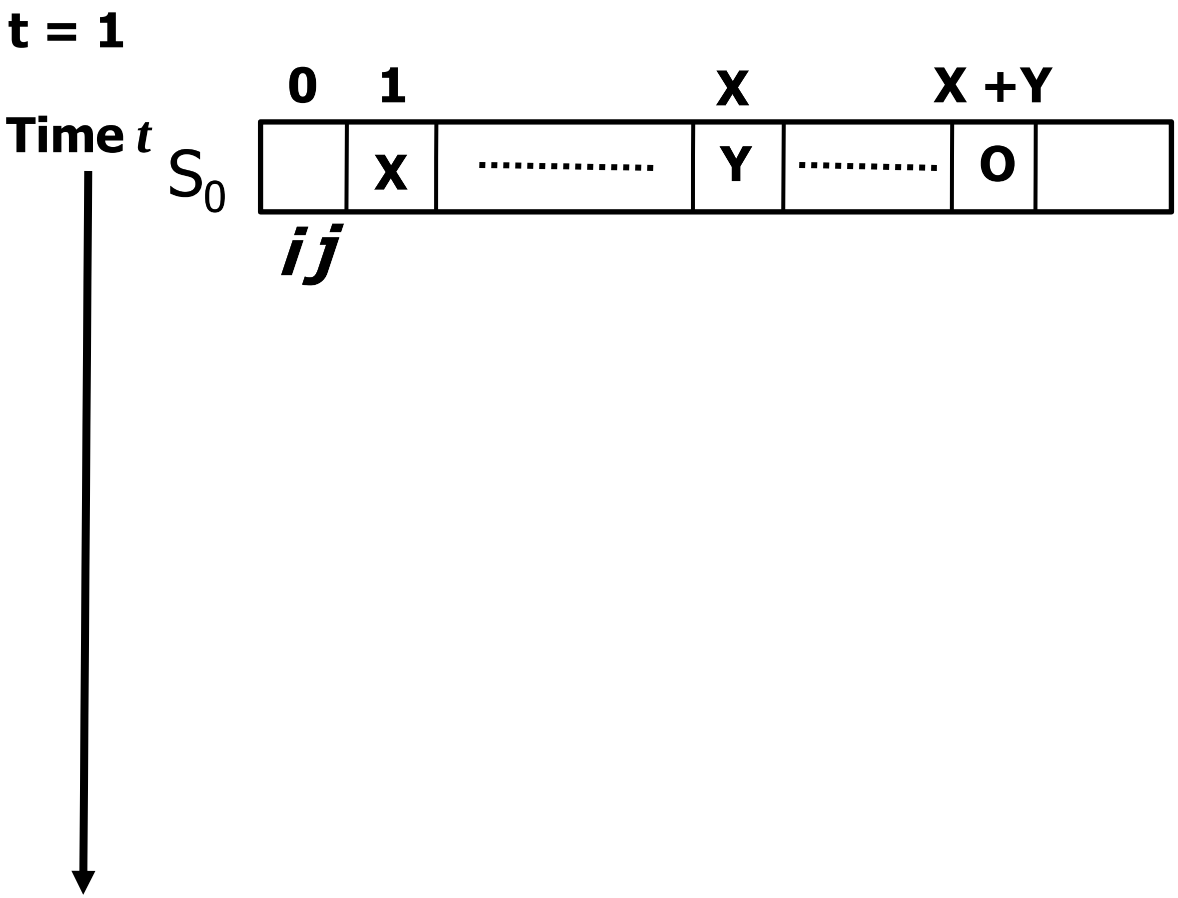
Key Scheduling Algorithm



i = 0
J = 0
$S_0[x] = x$
loop
 j = j +S[i] + K[i]
 swap(S[i], S[j])
 i = i + 1
end loop

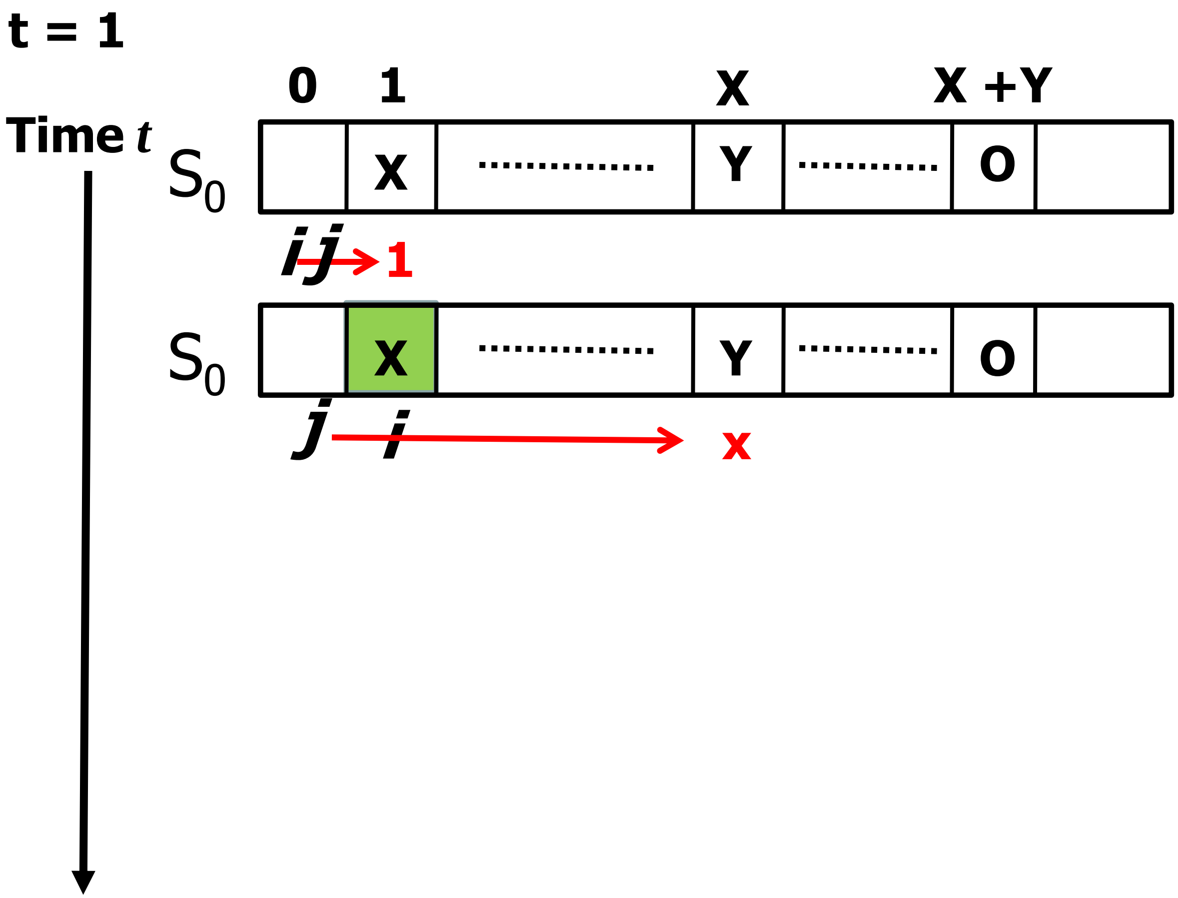
Initial state of PRGA

Pseudo Random Generator Algorithm



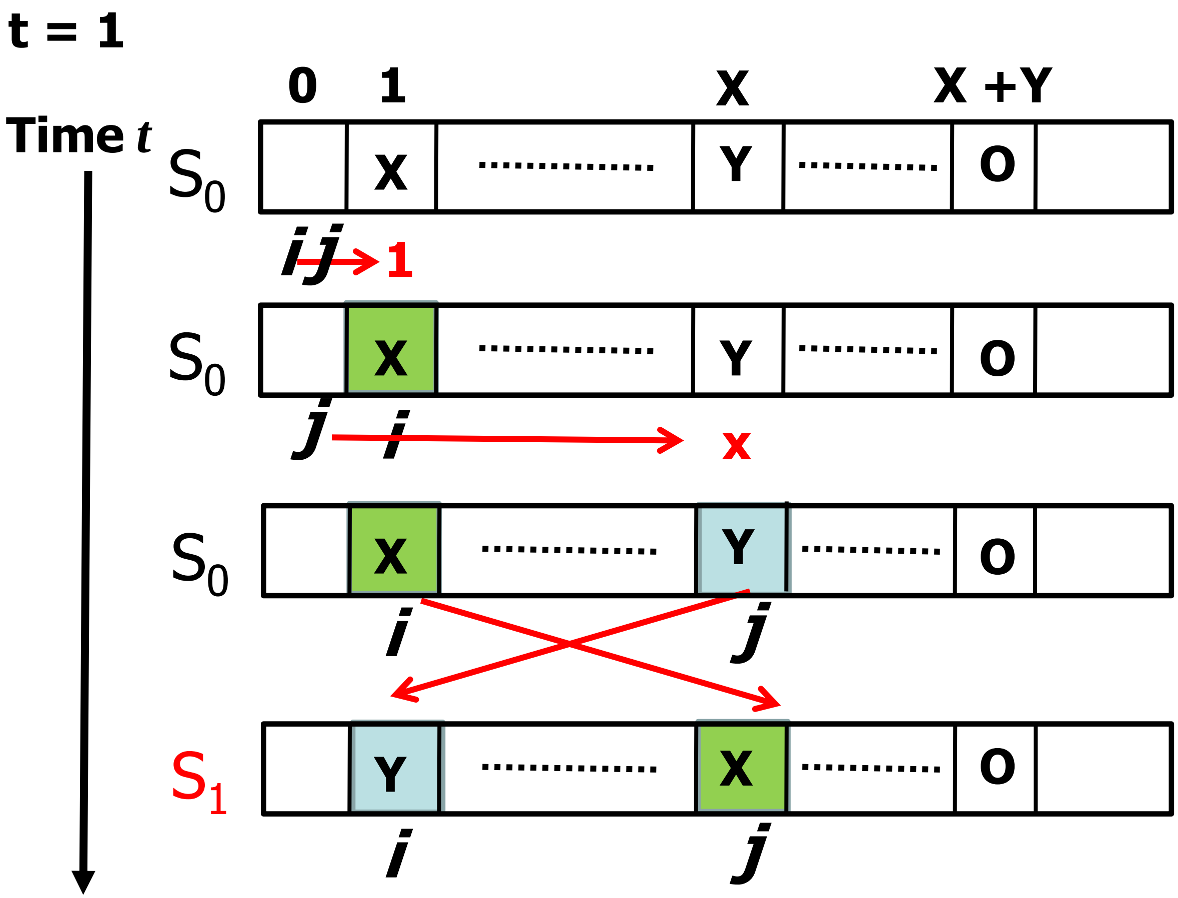

```
i = 0
J = 0
Loop
  i = i + 1
  j = j +S[ i ]
  swap(S[ i ], S[ j ])
  Z = S[S[ i ]+S[ j ]]
end loop
```

Pseudo Random Generator Algorithm



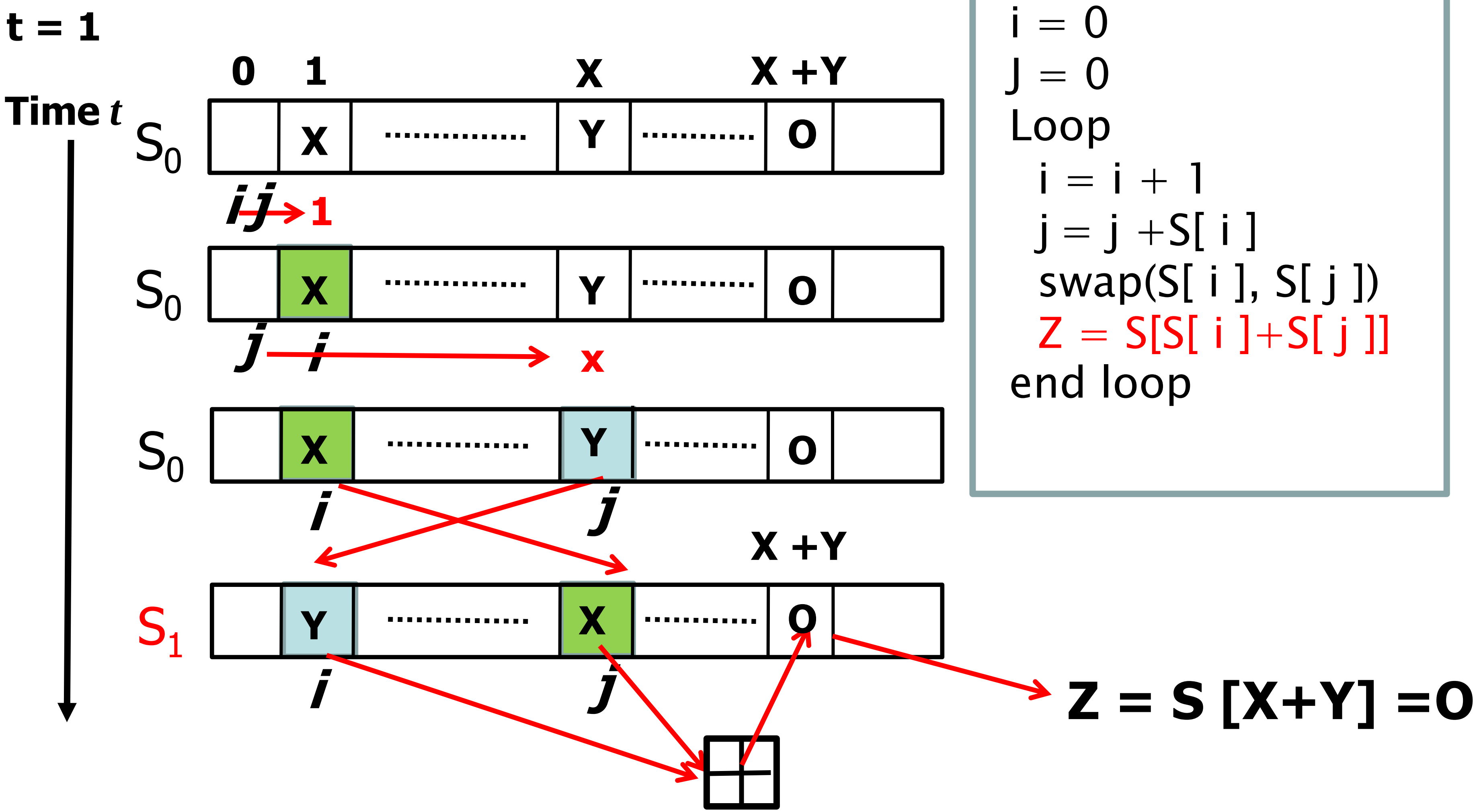

```
i = 0
J = 0
Loop
  i = i + 1
  j = j +S[ i ]
  swap(S[ i ], S[ j ])
  Z = S[S[ i ]+S[ j ]]
end loop
```

Pseudo Random Generator Algorithm




```
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
```

Pseudo Random Generator Algorithm




```
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
```

Known Results

Over the past 20 years, a number of results were published!

- State Recovery Attacks [KMPRV98, MK08]
- Distinguish Attacks [FM00, M'05, SVV10, SMPS12]
- Plaintext Recovery Attacks [MS01, MPS11]
- Other Attacks
  - Key Collision [M'09, JM11]
  - Key Recovery from Internal State [SM07, BC08]
  - Weak Keys [R'98]
  - Related Key Attack [JM12]

And More !

Initial Keystream Biases of RC4 and Its Applications (From FSE 2013)

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

-T. Isobe, T. Ohigashi, Y. Watanabe, M. Morii "Comprehensive Analysis of Initial Keystream Biases of RC4", IEICE Journal, to appear

Summary of Our Results

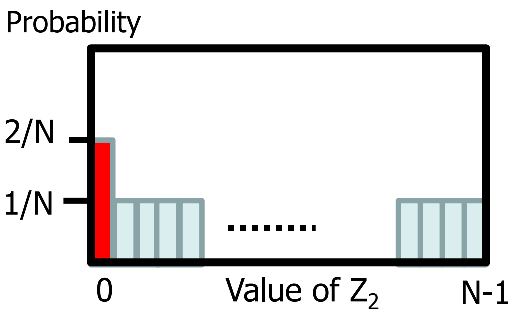
Comprehensively analyze initial biases of keystream
=> Find several new biases

- Theoretical : Prove "Why such biases occur in RC4?"
- Experimental : 2^{40} independent key test

- Applications
  - Plaintext Recovery Attack [FSE 2013]
  - Distinguishing Attack [IEICE]
  - Key (State) Recovery Attack [IEICE]

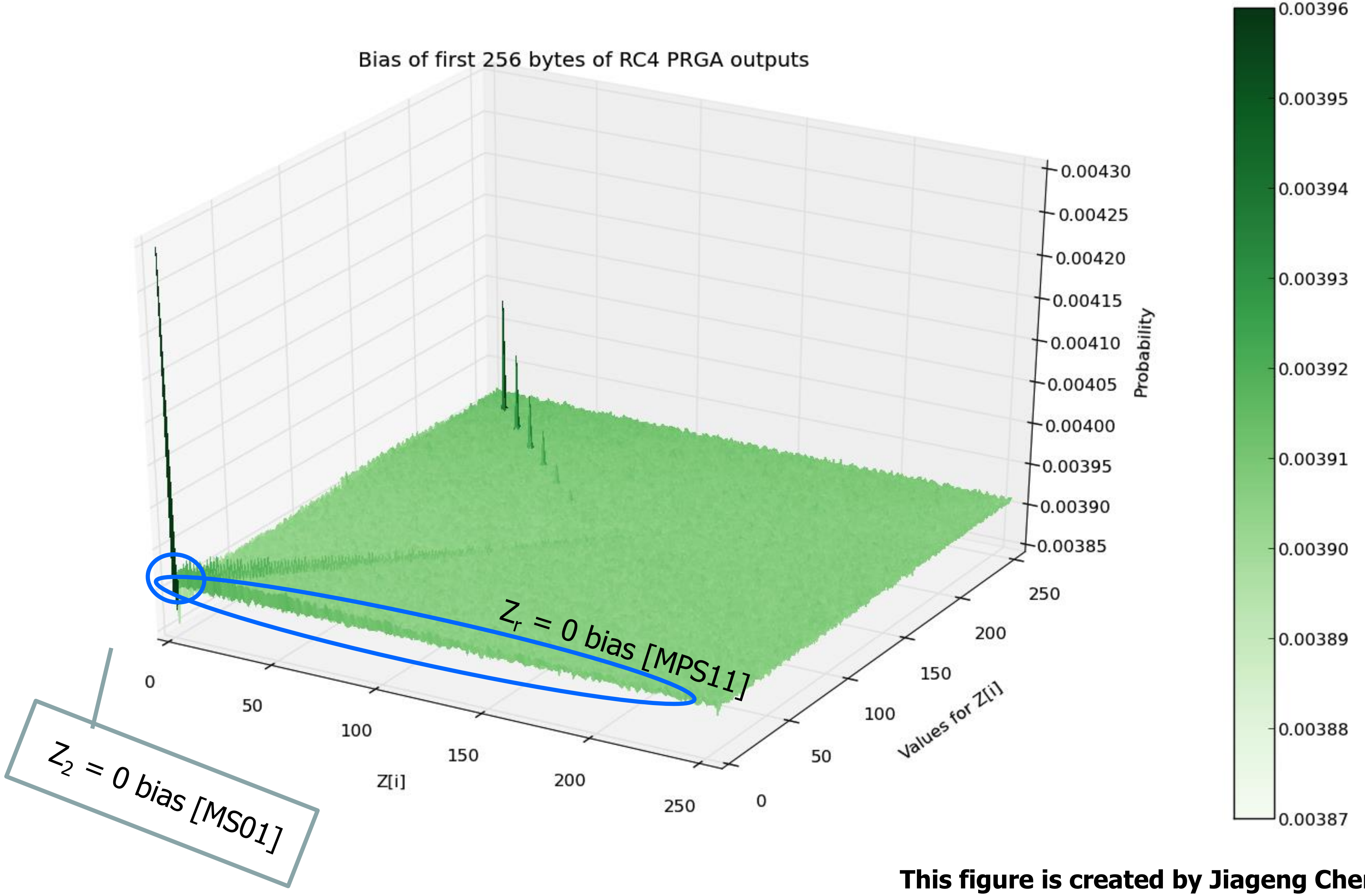
Known Biases of Initial Keystream

- 1st byte bias
  - Not uniform distribution
    - Experimentally found [Mir02]
    - Theoretically proofs [SMPS13]
- 2nd byte biases [MS01]
  - Strongly Biased to "0"



- 3rd to 255th byte biases [MPS11]
  - Biased to "0"

3D View of Keystrem Biases

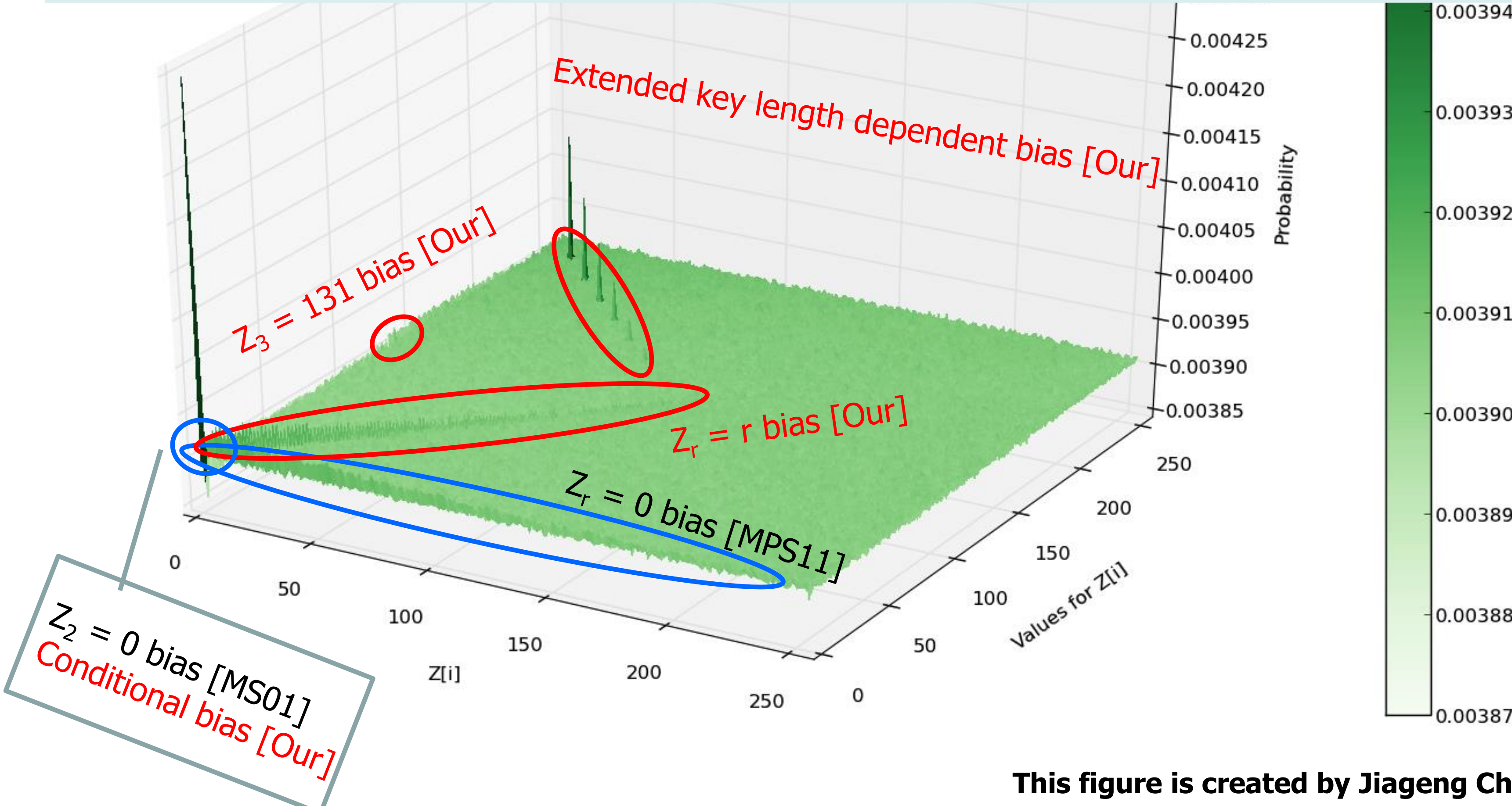


This figure is created by Jiageng Chen

New biases

Our Results

Find four types of new biases, and give theoretical reasons.
(Recent results [A+13] only shows experimental results)



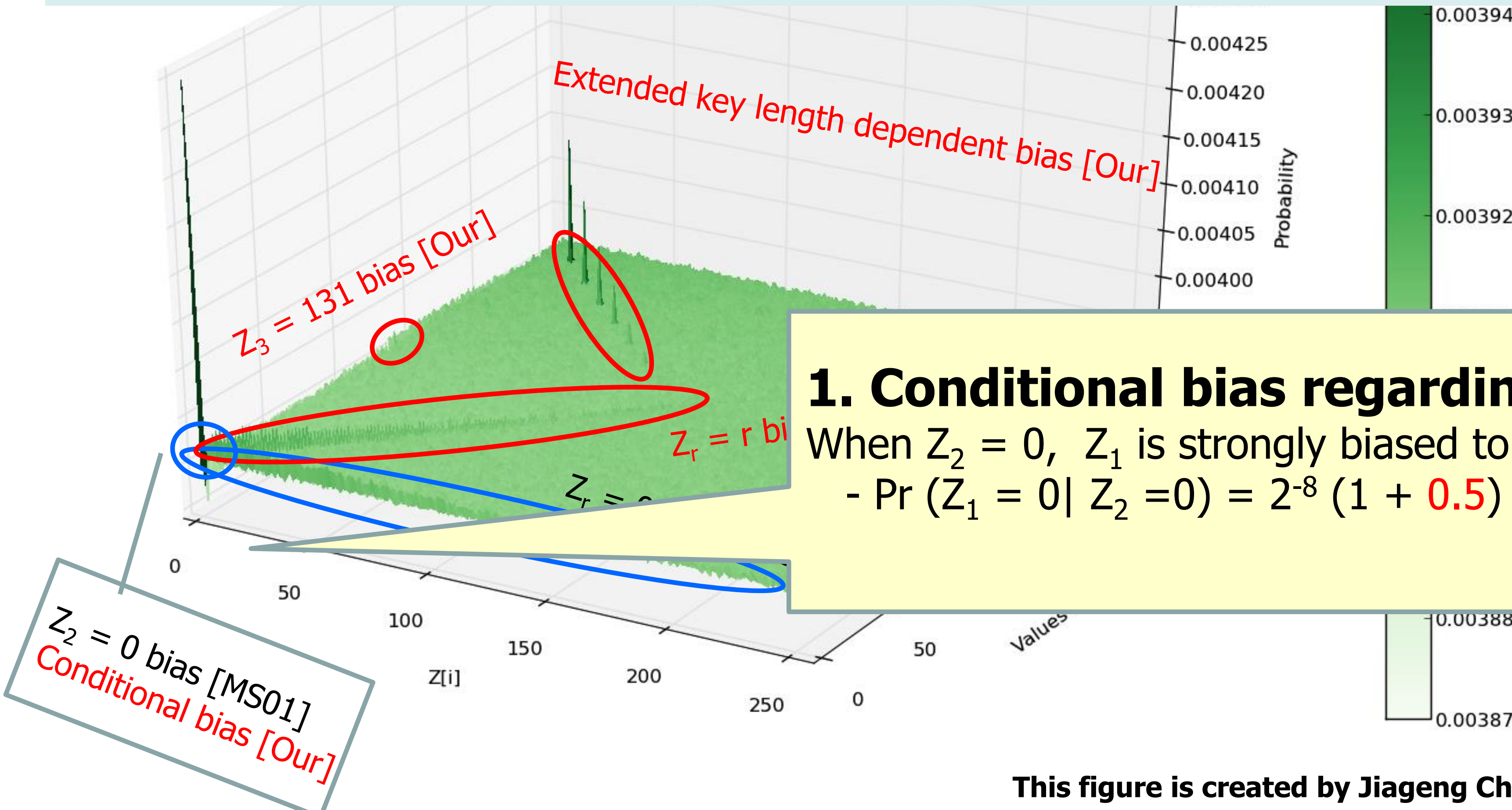
This figure is created by Jiageng Chen

New biases

Our Results

Find four types of new biases, and give theoretical reasons.

(Recent results [A+13] only shows experimental results)



1. Conditional bias regarding Z_1

When $Z_2 = 0$, Z_1 is strongly biased to "0"

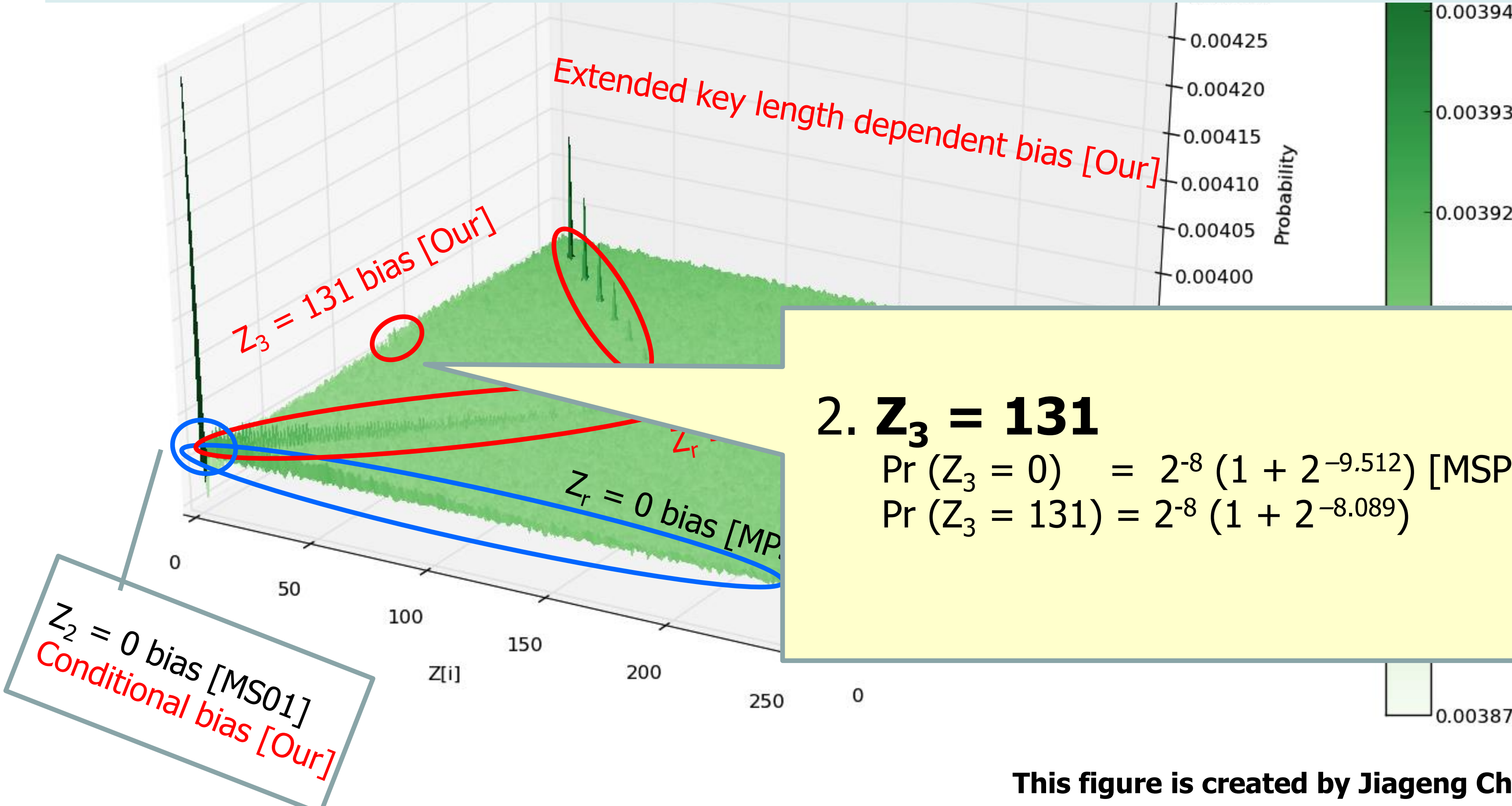
- $\Pr(Z_1 = 0 | Z_2 = 0) = 2^{-8}(1 + 0.5)$

This figure is created by Jiageng Chen

New biases

Our Results

Find four types of new biases, and give theoretical reasons.
(Recent results [A+13] only shows experimental results)



2. $\mathbf{Z_3 = 131}$

$\Pr(Z_3 = 0) = 2^{-8}(1 + 2^{-9.512})$ [MSP11]

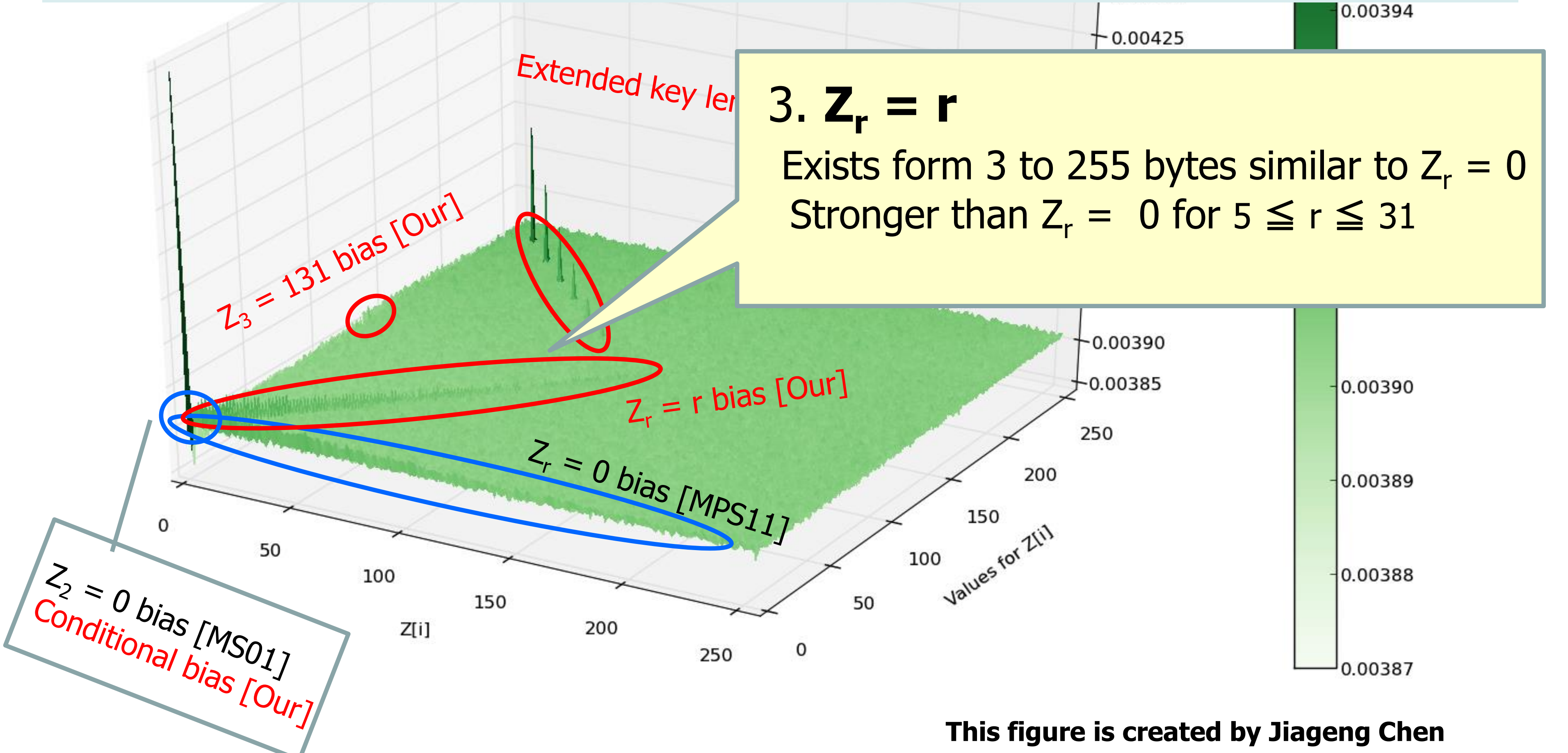
$\Pr(Z_3 = 131) = 2^{-8}(1 + 2^{-8.089})$

This figure is created by Jiageng Chen

New biases

Our Results

Find four types of new biases, and give theoretical reasons.
(Recent results [A+13] only shows experimental results)



3. Z_r = r

Exists form 3 to 255 bytes similar to $Z_r = 0$

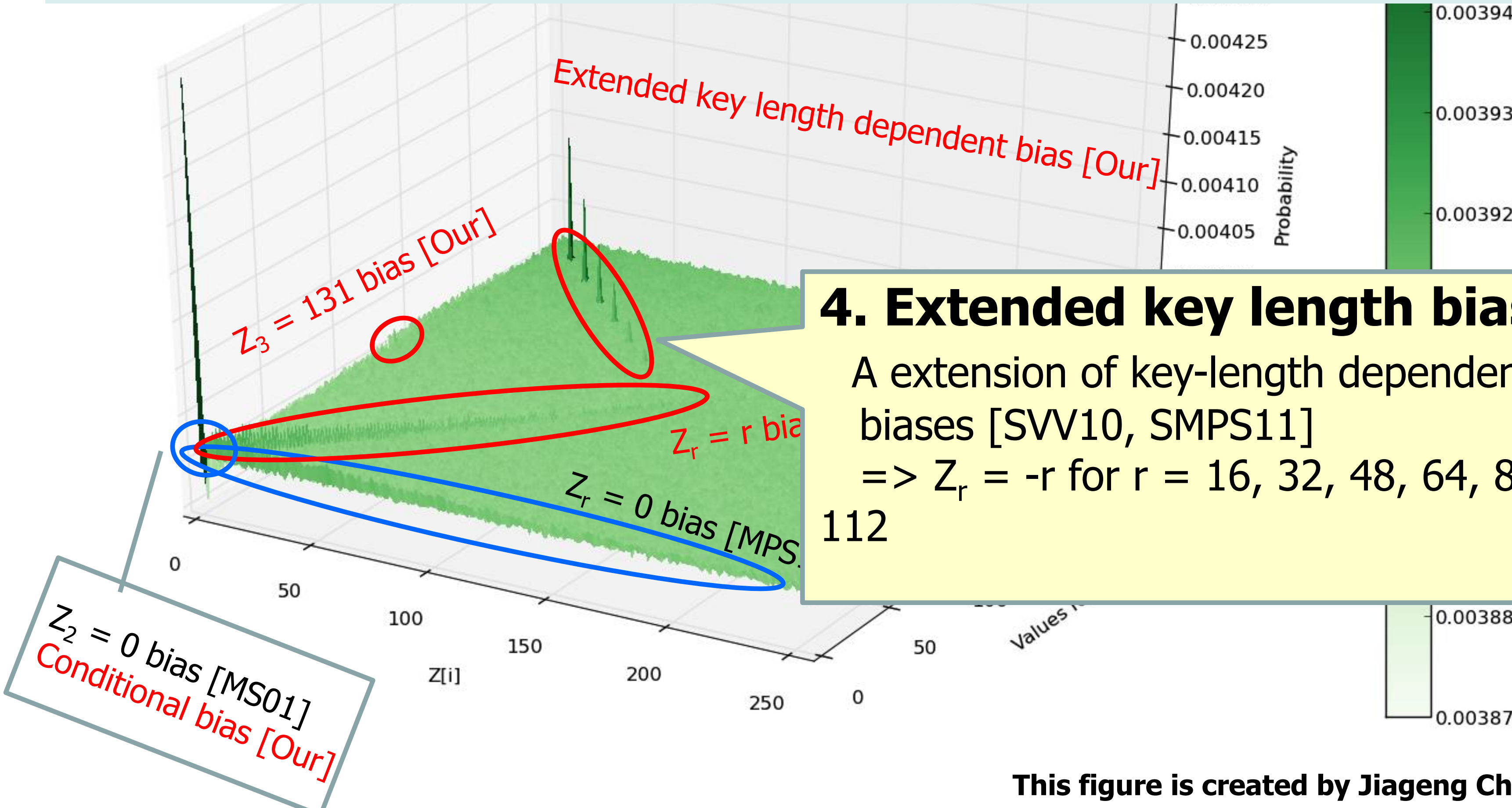
Stronger than $Z_r = 0$ for $5 \leqq r \leqq 31$

This figure is created by Jiageng Chen

New biases

Our Results

Find four types of new biases, and give theoretical reasons.
(Recent results [A+13] only shows experimental results)



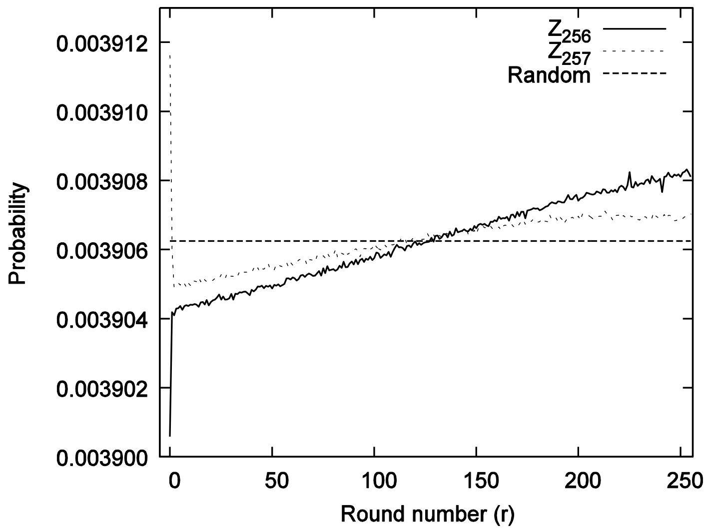
4. Extended key length bias

A extension of key-length dependent biases [SVV10, SMPS11]
=> $Z_r = -r$ for r = 16, 32, 48, 64, 80, 96, 112

This figure is created by Jiageng Chen

Other New Biases

- Experimentally found other two types of biases
  - **$Z_{256} = 0$**
    - Negative biases
    - $\Pr(Z_{256} = 0) = 2^{-8} \cdot (1 - 2^{-9.407})$
  - **$Z_{257} = 0$**
    - $\Pr(Z_{256} = 0) = 2^{-8} \cdot (1 + 2^{-9.531})$



However, no theoretical reason…orz

Recently these biases are proved
by Sarkar, Sen Gupta, Paul and Maitra [SSPM13]

Strongest Single–byte Biases

- List of strongest single-byte biases in first 257 bytes

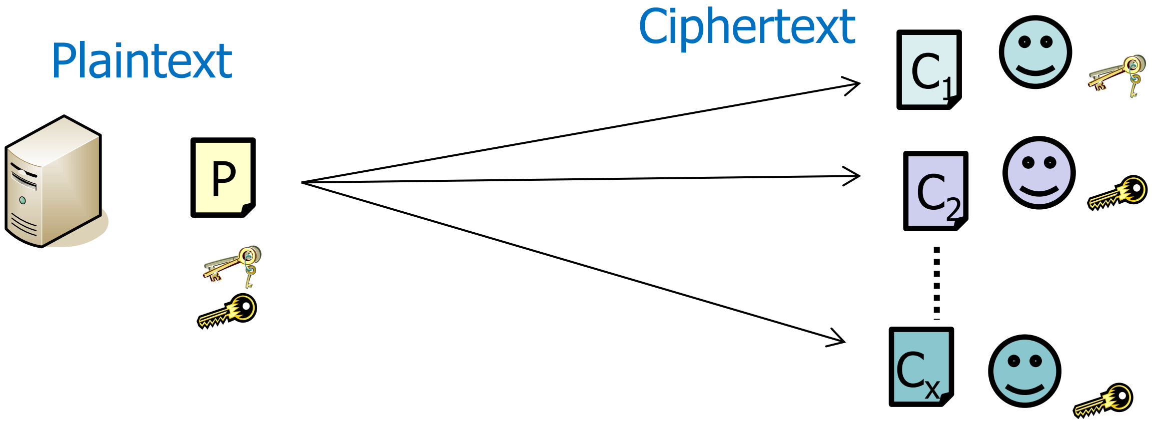
Table 1 Strongest single-byte set of first 257 bytes for $N = 256$ and $\ell = 16$

r	Strongest single-byte bias	Prob.(Theoretical)	Prob.(Experimental)
1	$Z_1 = 129$ (negative bias) [1]	N/A	$2^{-8} \cdot (1 - 2^{-7.214})$
2	$Z_2 = 0$ [12]	$2^{-8} \cdot (1 + 2^{0})$	$2^{-8} \cdot (1 + 2^{0.002})$
3	$Z_3 = 131$ (Our)	$2^{-8} \cdot (1 + 2^{-8.089})$	$2^{-8} \cdot (1 + 2^{-8.109})$
4	$Z_4 = 0$ [9]	$2^{-8} \cdot (1 + 2^{-7.581})$	$2^{-8} \cdot (1 + 2^{-7.611})$
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]	$2^{-8} \cdot (1 + 2^{-4.841})$	$2^{-8} \cdot (1 + 2^{-4.811})$
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)	$2^{-8} \cdot (1 + 2^{-5.404})$	$2^{-8} \cdot (1 + 2^{-5.383})$
33–47	$Z_r = 0$ [9]	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)	$2^{-8} \cdot (1 + 2^{-5.981})$	$2^{-8} \cdot (1 + 2^{-5.938})$
49–63	$Z_r = 0$ [9]	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)	$2^{-8} \cdot (1 + 2^{-6.576})$	$2^{-8} \cdot (1 + 2^{-6.496})$
65–79	$Z_r = 0$ [9]	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)	$2^{-8} \cdot (1 + 2^{-7.192})$	$2^{-8} \cdot (1 + 2^{-7.224})$
81–95	$Z_r = 0$ [9]	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)	$2^{-8} \cdot (1 + 2^{-7.831})$	$2^{-8} \cdot (1 + 2^{-7.911})$
97–111	$Z_r = 0$ [9]	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)	$2^{-8} \cdot (1 + 2^{-8.500})$	$2^{-8} \cdot (1 + 2^{-8.666})$
113–255	$Z_r = 0$ [9]	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	$Z_{256} = 0$ (negative bias) (Our)	N/A	$2^{-8} \cdot (1 - 2^{-9.407})$
257	$Z_{257} = 0$ (Our)	N/A	$2^{-8} \cdot (1 + 2^{-9.531})$

Applications to
Plaintext Recovery Attack

Plaintext Recovery in Broadcast Setting

- Broadcast setting
  - Same plaintext is encrypted with different (user) keys



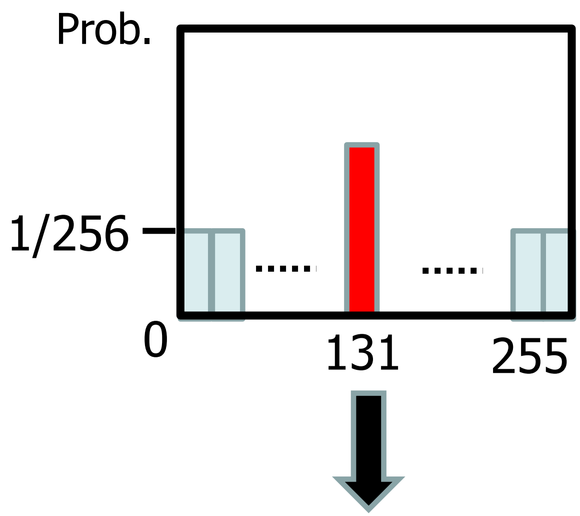
- Plaintext Recovery Attack
  - Extract plaintext from ONLY ciphertexts encrypted by different keys
  - Passive attack
    - What attacker should do is to collect ciphertexts
    - NOT use additional information such as timing and delays.

Idea for Plaintext Recovery Attack [MS 01]

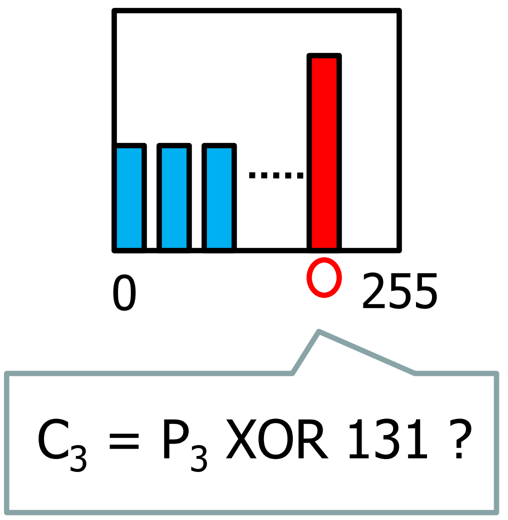
- **Relation in each byte**

  => "$C_i = P_i$ XOR Z_i"

  - If P_i is fix, the distribution of Z_i maps to C_i
  - If $Z_3 = 131$, then $C_3 = P_3$ XOR 131
  - Most frequent value of C_3 is P_3 XOR 131



Frequency Table of C_3

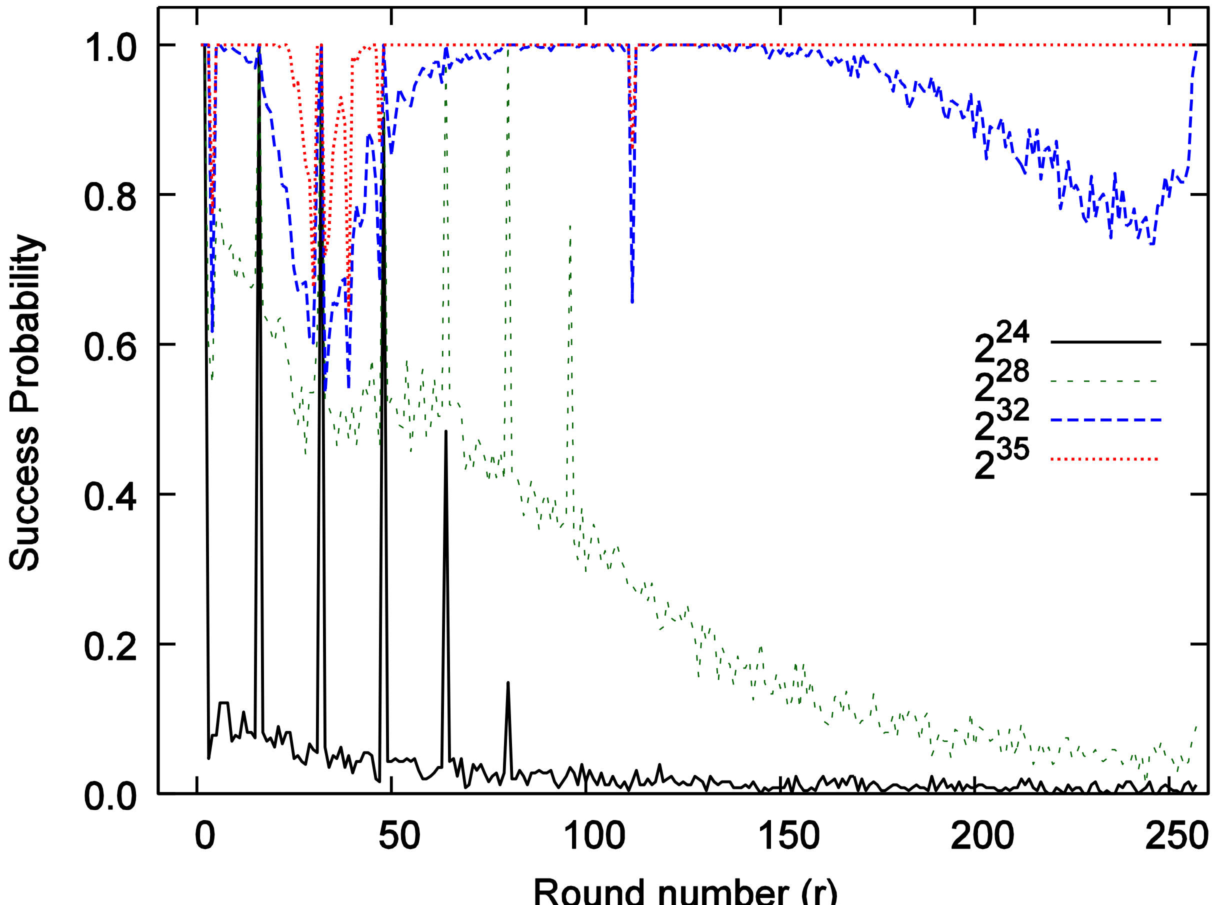


- **Algorithm** : Plaintext Recovery Attack
  1. Collect X ciphertexts $C^{(1)}, \ldots, C^{(X)}$
  2. Count the values of C_i and make a frequency table
  3. Regard Most frequent values of C_i as P_i XOR Z'_x

     Z'x : strongest biased value in our table.

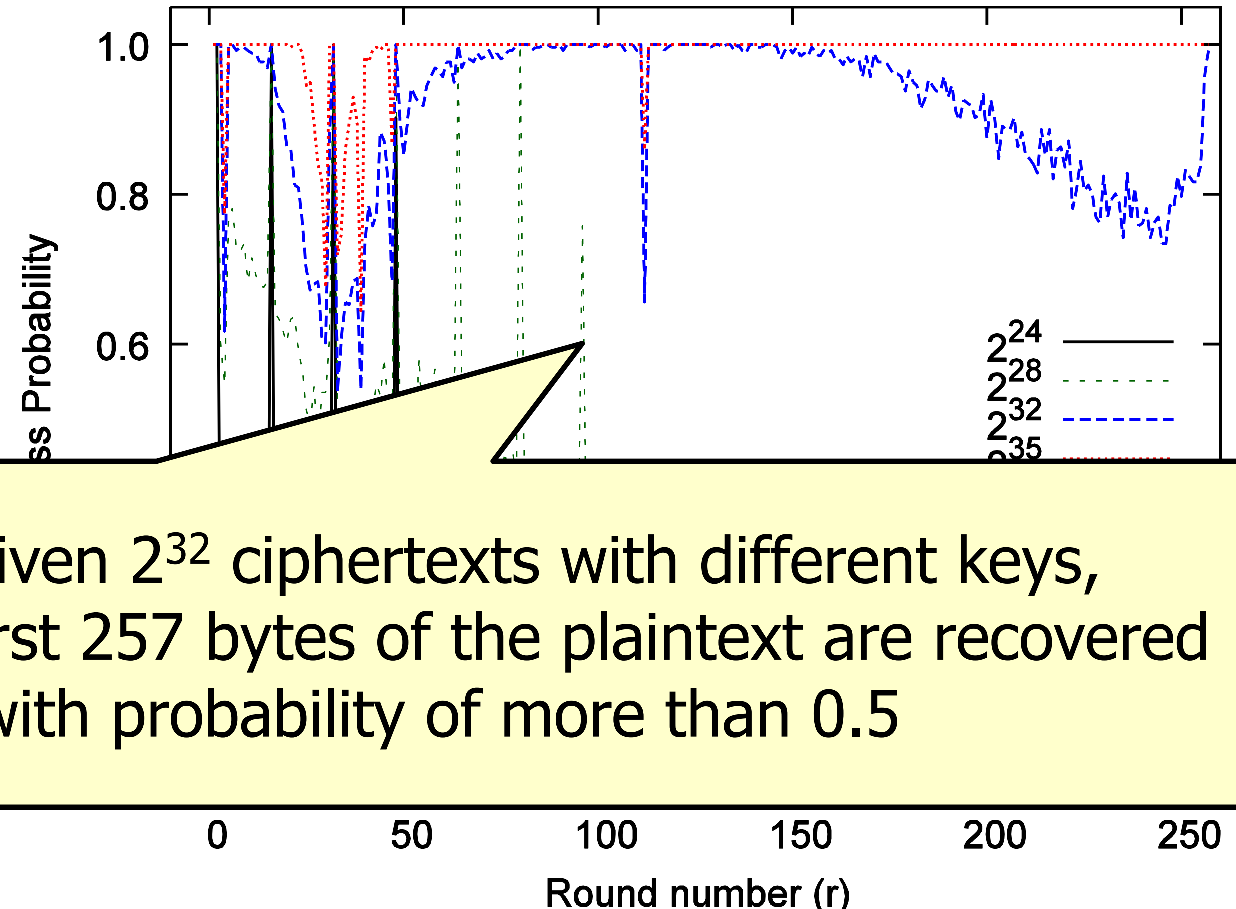
Experimental Results

- Experiment for 256 different plaintexts in the cases where $2^6, \ldots, 2^{35}$ ciphertexts with randomly-chosen keys are given.



Experimental Results

- Experiment for 256 different plaintexts in the cases where $2^6, . . . , 2^{35}$ ciphertexts with randomly-chosen keys are given.



Given 2^{32} ciphertexts with different keys, first 257 bytes of the plaintext are recovered with probability of more than 0.5

Other Plaintext Recovery Attack

How to Recover later byte (after 258 bytes)?

Plaintext Recovery Attack
Using Initial Biases

P_1, P_2, , …, P_{256}, P_{257},

？？？

P_{258}, P_{259} …

- Use Mantin's long term bias
  - Occur any bytes of a keystream

Mantin's Long Term Biases

- Digraph Repetition Bias
  - Known strongest long term bias
  - Same pattern appear after G bytes

Key Stream ....ABHLWECTSDGAB....

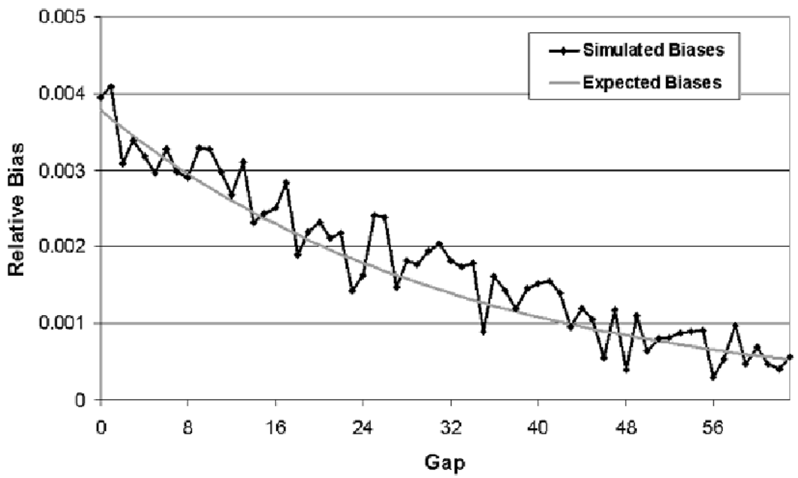
gap *G*

G=0
G=1
G=2
⋮
G=253

strong ↑ bias ↓ weak

$$Z_t \,||\, Z_{t+1} = Z_{t+2+G} \,||\, Z_{t+3+G}$$

Probability (ideal) : $1/N^2$
Probability (RC4) : $1/N^2(1 + p)$



Our Method

- Relation for plaintext recovery attacks

$$
\begin{aligned}
&(C_r \mid\mid C_{r+1}) \oplus (C_{r+2+G} \mid\mid C_{r+3+G}) \\
&= (P_r \oplus Z_r \mid\mid P_{r+1} \oplus Z_{r+1}) \oplus (P_{r+2+G} \oplus Z_{r+2+G} \mid\mid P_{r+3+G} \oplus Z_{r+3+G}) \\
&= (P_r \oplus P_{r+2+G} \oplus Z_r \oplus Z_{r+2+G} \mid\mid P_{r+1} \oplus P_{r+3+G} \oplus Z_{r+1} \oplus Z_{r+3+G}).
\end{aligned}
$$

Our Method

- Relation for plaintext recovery attacks

$$
\begin{aligned}
&(C_r \mid\mid C_{r+1}) \oplus (C_{r+2+G} \mid\mid C_{r+3+G}) \\
&= (P_r \oplus Z_r \mid\mid P_{r+1} \oplus Z_{r+1}) \oplus (P_{r+2+G} \oplus Z_{r+2+G} \mid\mid P_{r+3+G} \oplus Z_{r+3+G}) \\
&= (P_r \oplus P_{r+2+G} \oplus \underline{Z_r \oplus Z_{r+2+G}} \mid\mid P_{r+1} \oplus P_{r+3+G} \oplus \underline{Z_{r+1} \oplus Z_{r+3+G}}).
\end{aligned}
$$

Assuming $Z_t \mid\mid Z_{t+1} = Z_{t+2+G} \mid\mid Z_{t+3+G}$, (Mantin's relation)

Our Method

- Relation for plaintext recovery attacks

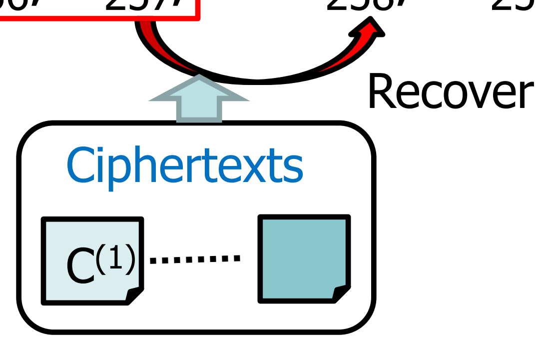
$$(C_r \mid\mid C_{r+1}) \oplus (C_{r+2+G} \mid\mid C_{r+3+G})$$
$$= (P_r \oplus Z_r \mid\mid P_{r+1} \oplus Z_{r+1}) \oplus (P_{r+2+G} \oplus Z_{r+2+G} \mid\mid P_{r+3+G} \oplus Z_{r+3+G})$$
$$= (P_r \oplus P_{r+2+G} \oplus \underline{Z_r \oplus Z_{r+2+G}} \mid\mid P_{r+1} \oplus P_{r+3+G} \oplus \underline{Z_{r+1} \oplus Z_{r+3+G}}).$$

Assuming $Z_t \mid\mid Z_{t+1} = Z_{t+2+G} \mid\mid Z_{t+3+G}$, (Mantin's relation)

$(C^r \mid\mid C^{r+1})\ \ (C_{r+2+G} \mid\mid C_{r+3+G}) = (P_r \mid\mid P_{r+2})\ \ (P_{r+1} \mid\mid P_{r+3+G})$

$P_1, P_2, , \ldots, P_{256}, P_{257},$ P_{258}, P_{259}

Know bytes



Guess by using long term bias with parameters G = 0, 1, …66

Experimental Results

- Experimental
  - $P_{258}, .., P_{261}$ are recovered from 2^{34} ciphertexts

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

- Theoretical
  - Given 2^{34} ciphertexts with different keys, 1000 TB bytes of plaintext are recovered with probability of 0.99

Advanced Plaintext Recovery Attacks on RC4 (From SAC 2013)

-T. Ohigashi, T. Isobe, Y. Watanabe, M. Morii "How to Recover Any Bytes of Plaintext on RC4", SAC 2013

Overview

- Previous Plaintext Recovery Attack (FSE 2013)
  - Exploit biases in initial bytes of keystream

Plaintext Recovery Attack
Using Initial Bias

$P_1, P_2, , \ldots, P_{256}, P_{257},$ $\quad$ $P_{258}, P_{259} \quad \cdots$

Mantin Bias

  - If first bytes are disregarded, it seems to be secure
  - Countermeasure : RC4 –drop(n)

First n bytes are disregard

Key → RC4 → $Z_1, Z_2, \ldots, Z_n, Z_{n+1}, \ldots,$

Advanced Plaintext Recovery Attacks

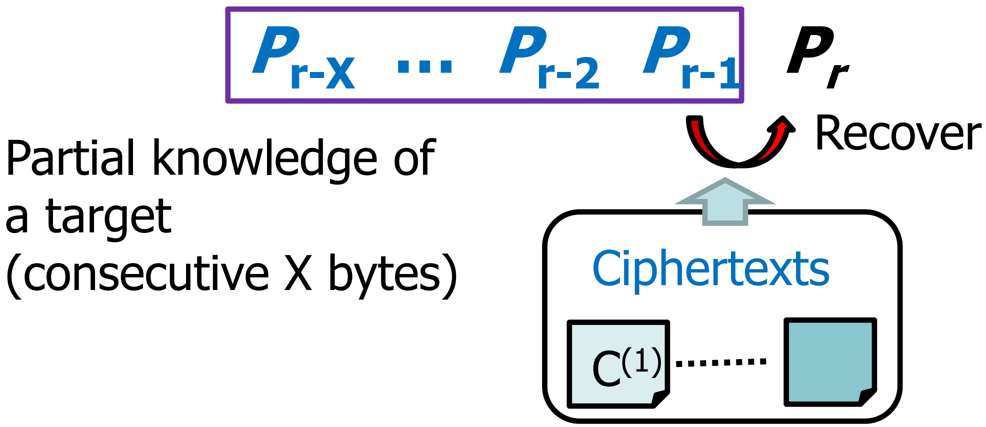
Two types of plaintext recovery attacks on RC4-drop

- Method 1 : Modified FSE 2013 Attack
  - Use partial knowledge of a plaintext
  - Works even if first bytes are disregarded
- Method 2: Guess and Determine Plaintext Recover Attack
  - Combine use of two types of long term biases
  - Do not require any knowledge of plaintext

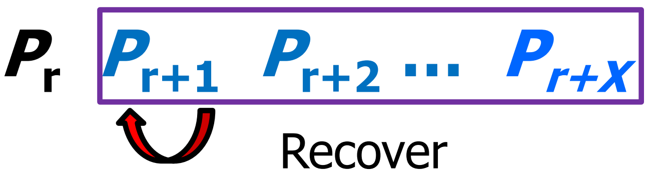
Method 1: Plaintext Recovery Attack using Known Partial Plaintext Bytes

- Simple extension of FSE 2013 attack
  - generalize FSE 2013's attack functions based on Mantin's biases
  - Use Mantin bias with partial knowledge in forward and backward manner

Forward attack function

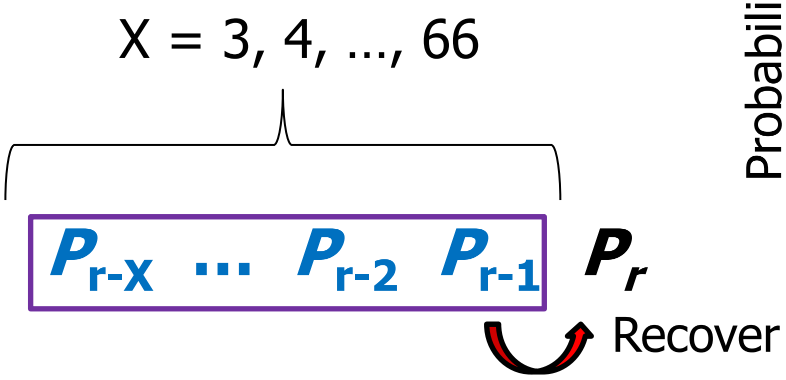


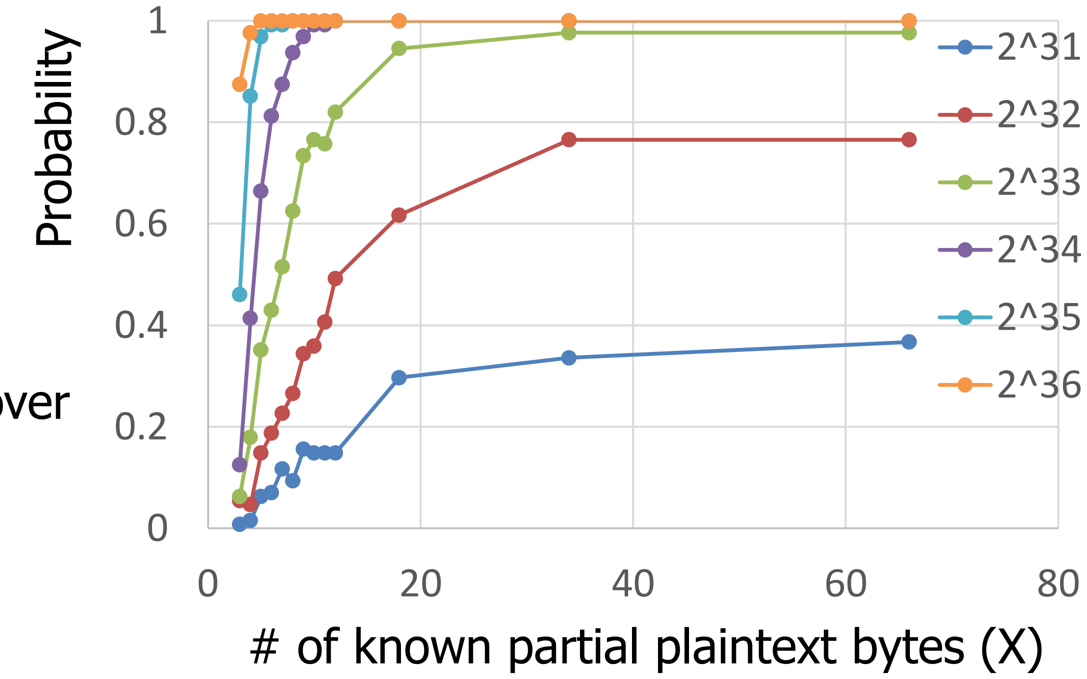
Backward attack function



Experimental Results

- Probability for recovering the target byte, given X bytes of knowledge of the plaintext



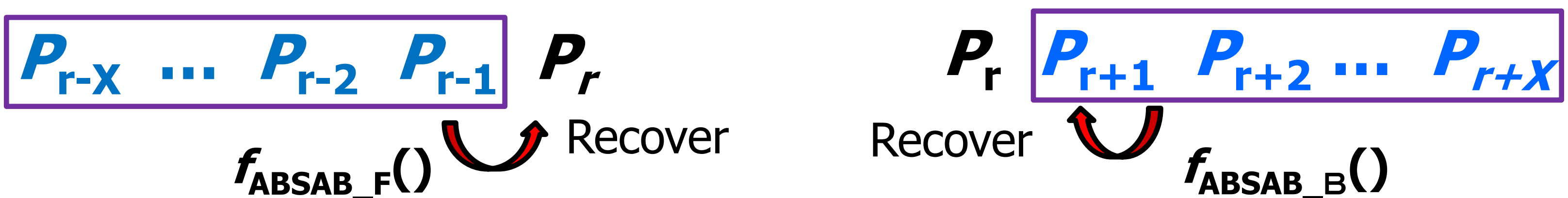


ex.) Given only 6 bytes of knowledge of a plaintext, other bytes are recovered with 2^{34} ciphertexts

Method 2: Guess and Determine Plaintext Recover Attack

- Based on two types of long-term biases
  - Mantin's long-term bias (ABSAB bias)
  - Fluhrer-McGrew bias in FSE 2000 (FM00 bias)

Attack function based on ABSAB bias (the same as the first attack)



Attack function based on FM00 bias (NEW)
(FM00 bias : 2-byte conditional bias)

Attack Procedure

1. Guess the value of P_r

P_r Target byte

Step 1:
Set a candidate of *P_r*

Attack Procedure

1. Guess the value of P_r
2. Guess X bytes of the plaintext from P_r (guessed in Step 1) by FM00 bias

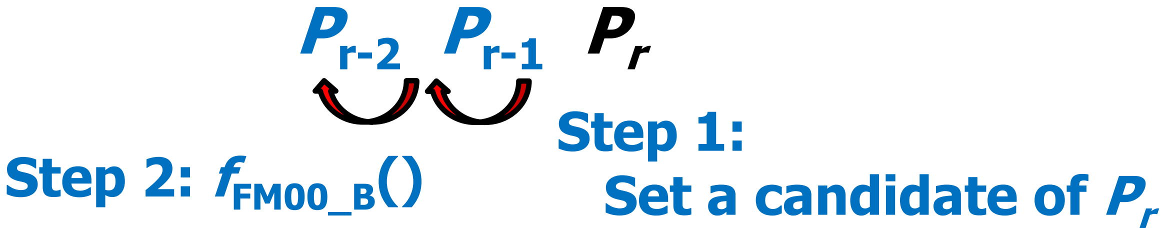
P_{r-1} P_r

Step 1:
Set a candidate of P_r

Step 2: $f_{FM00\_B}()$

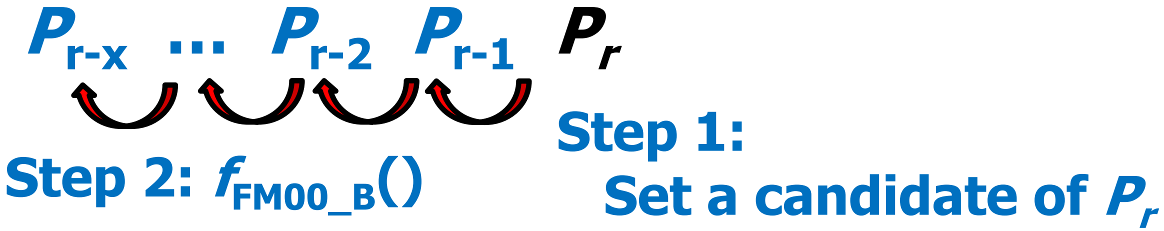
Attack Procedure

1. Guess the value of P_r
2. Guess X bytes of the plaintext from P_r (guessed in Step 1) by FM00 bias



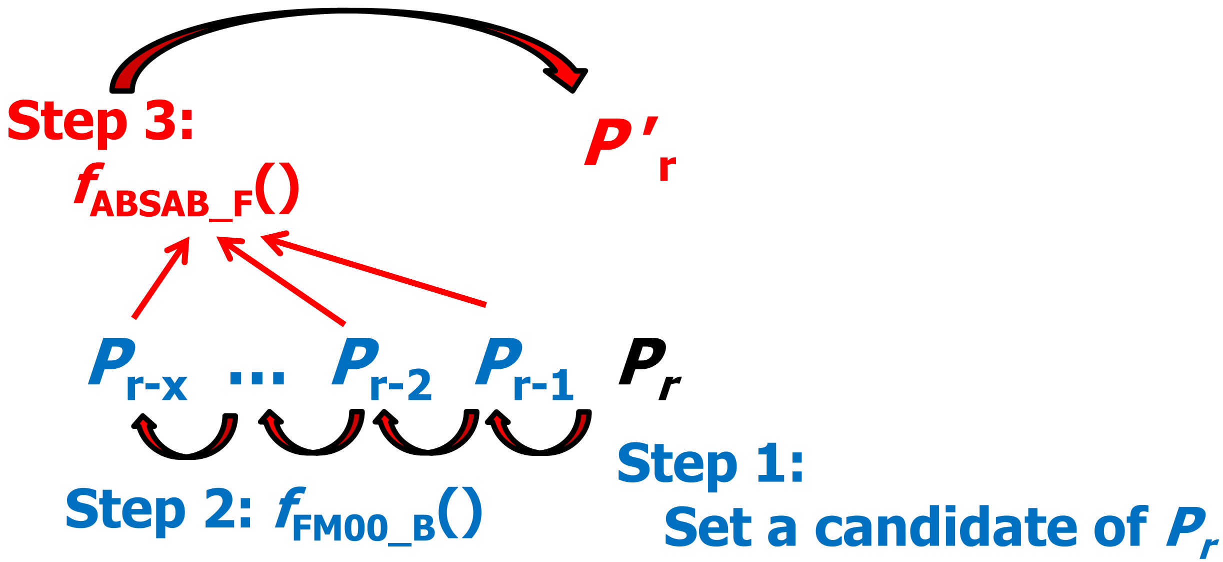
Attack Procedure

1. Guess the value of P_r
2. Guess X bytes of the plaintext from P_r (guessed in Step 1) by FM00 bias



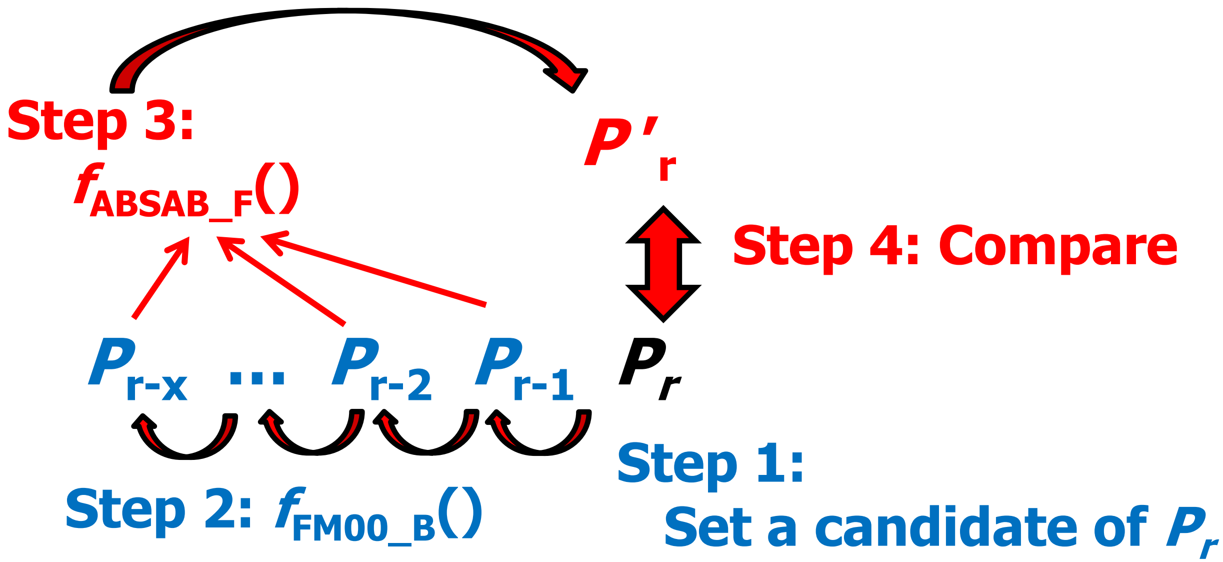
Attack Procedure

1. Guess the value of P_r
2. Guess X bytes of the plaintext from P_r (guessed in Step 1) by FM00 bias
3. Guess P'_r from P_{r-x}, …, P_{r-1} (guessed in Step 2) by ABSAB bias



Attack Procedure

1. Guess the value of P_r
2. Guess X bytes of the plaintext from P_r (guessed in Step 1) by FM00 bias
3. Guess P'_r from P_{r-x}, …, P_{r-1} (guessed in Step 2) by ABSAB bias
4. If P'_r is not equal to P_r guessed in Step 1, the value is wrong.
   Otherwise the value is regarded as a candidate of correct P_r



Experimental Results

- Probability for recovering a byte of a plaintext on RC4-drop(3072)
  - Obtained from 256 test
  - # of ciphertexts: 2^{32}, 2^{33}, 2^{34}, 2^{35}
  - Target Plaintext byte in this experiment: P_{128}

	# of ciphertexts			
	2^{32}	2^{33}	2^{34}	2^{35}
P_{128}	0.0039	0.1133	0.9102	1.0000

- **Given 2^{35} ciphertexts,**
  => recover any plaintext byte with probability close to one
- **Given 2^{34} ciphertexts,**
  =>recover any plaintext byte with probability of about 0.91

Conclusion

This talk introduced two recent results on RC4

-Initial Keystream Biases of RC4 and Its Applications

(From FSE 2013 and IEICE Journal)

-Advanced Plaintext Recovery Attacks on RC4-drop

(From SAC 2013)