HIROSHIMA UNIVERSITY

15 August, 2013
SAC 2013 @ Simon Fraser University

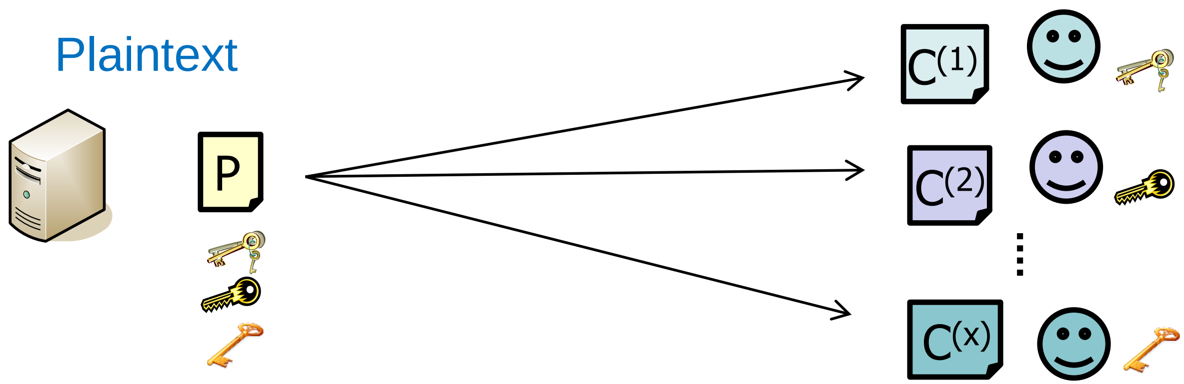
KOBE UNIVERSITY

How to Recover Any Byte of Plaintext on RC4

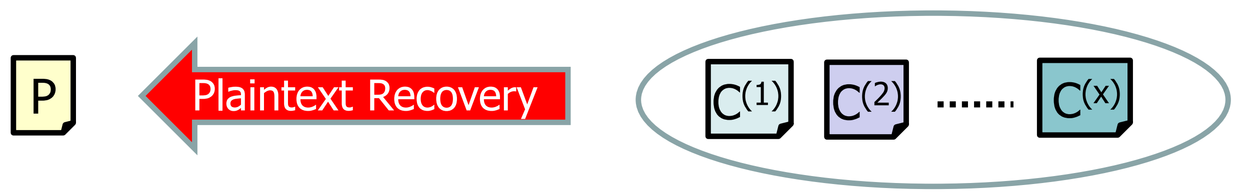
Toshihiro Ohigashi (Hiroshima University)
Takanori Isobe (Kobe University)
Yuhei Watanabe (Kobe University)
Masakatu Morii (Kobe University)

Target

- Broadcast setting
  - Same plaintext is encrypted with different (user) keys (e.g. Group mail)
  - can be easily converted into the multi-session setting of SSL/TLS
    - Target plaintext blocks are repeatedly sent in the same position of plaintext



- Plaintext Recovery Attack in the broadcast/multi-session setting
  - Recover a plaintext from ONLY ciphertexts encrypted by different keys
  - Passive attack
    - What attacker should do is to collect ciphertexts
    - NOT use additional information such as side channel information



Related Works

- Plaintext Recovery Attack on (pure) RC4 in these settings
  - Mantin-Shamir Attack (FSE 2001)
    - recover 2^{nd} byte of a plaintext from $\Omega(N)$ ciphertexts with probability more than a random search, where $N = 256$
  - Maitra-Paul-SenGupta Attack (FSE 2011)
    - recover 3^{rd} to 255^{th} bytes of a plaintext from $\Omega(N^3)$ ciphertexts with probability more than a random search, where $N = 256$
  - Isobe-Ohigashi-Watanabe-Morii Attack (FSE 2013)
    - recover 1^{st} to 257^{th} bytes of a plaintext from 2^{32} ciphertexts with probability of > 0.5
    - recovery first 1 petabytes of a plaintext from 2^{34} ciphertexts with probability closed to one
  - AlFardan-Bernstein-Paterson-Poettering-Schuldt Attack (USENIX Security 2013, Aug. 15, 2013, Today !)
    - recover 1^{st} to 256^{th} bytes of a plaintext from 2^{32} ciphertexts with probability of > 0.96

Related Works

- Plaintext Recovery Attack on (pure) RC4 in these settings
  - Mantin-Shamir Attack (FSE 2001)
    - recover 2nd byte of a plaintext from $\Omega(N)$ ciphertexts with probability more than a random search, where $N = 256$
  - Maitra-Paul-SenGupta Attack (FSE 2011)
    - recover 3rd to 255th bytes of a plaintext from $\Omega(N^3)$ ciphertexts with probability more than a random search, where $N = 256$
  - Isobe-Ohigashi-Watanabe-Morii Attack (FSE 2013)
    - recover 1st to 257th bytes of a plaintext from 2^{32} ciphertexts with probability of > 0.5

But, these attacks do not work on a relatively secure implementation of RC4 (RC4-drop)

- disregards the first *n* bytes of a keystream of RC4

*** recommendation: *n*=512 or 768, (conservative) *n* = 3072 by Mironov in CRYPTO 2002**

HIROSHIMA UNIVERSITY

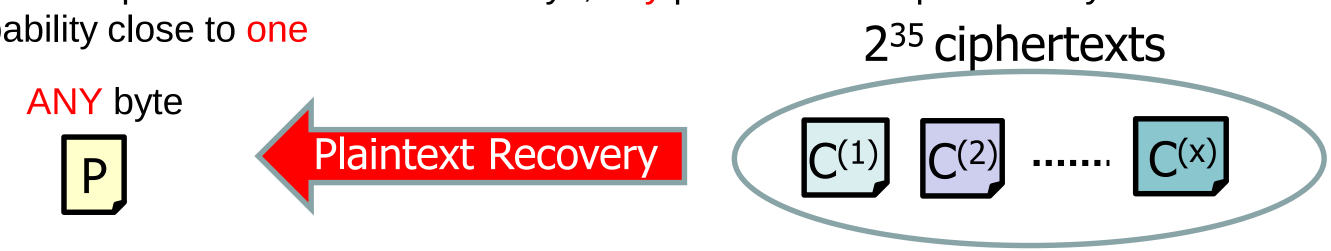
Summary of Our Results

Security Evaluation of RC4-drop in the Broadcast/Multi-session Setting

Results

- **Plaintext recovery attack using Known Partial Plaintext Bytes**
  - Based on Mantin's long-term bias in EUROCRYPT 2005
  - Given consecutive 6 bytes of a target plaintext and 2^{34} ciphertexts with different keys, consecutive 1 petabytes of the plaintext are recovered with probability more than 0.6

- **Guess-and-Determine Plaintext Recovery Attack**
  - Combine use of Mantin's long-term bias and Fluhrer-McGrew long-term bias in FSE 2000
  - Not Require any previous knowledge of a plaintext
  - Given **2^{35}** ciphertexts with different keys, any position of the plaintext byte is recovered with probability close to one



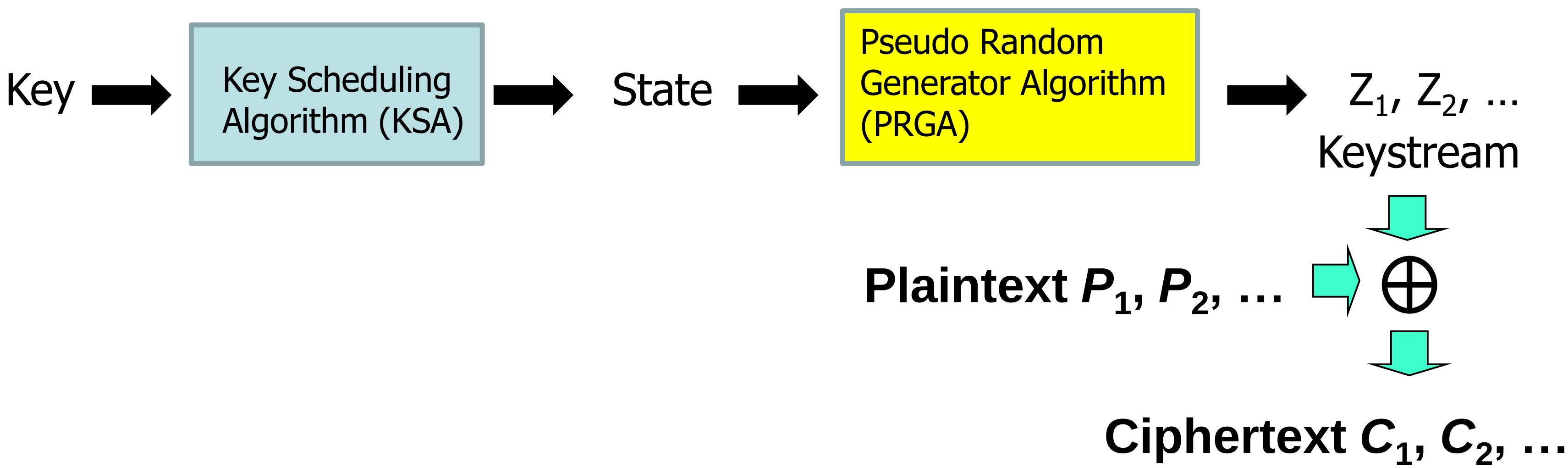
Agenda

- RC4 Stream Cipher
- Previous Plaintext Recovery Attacks
- Plaintext Recovery Attack using Known Partial Plaintext Bytes
- Guess-and-Determine Plaintext Recovery Attack
- Conclusion

RC4

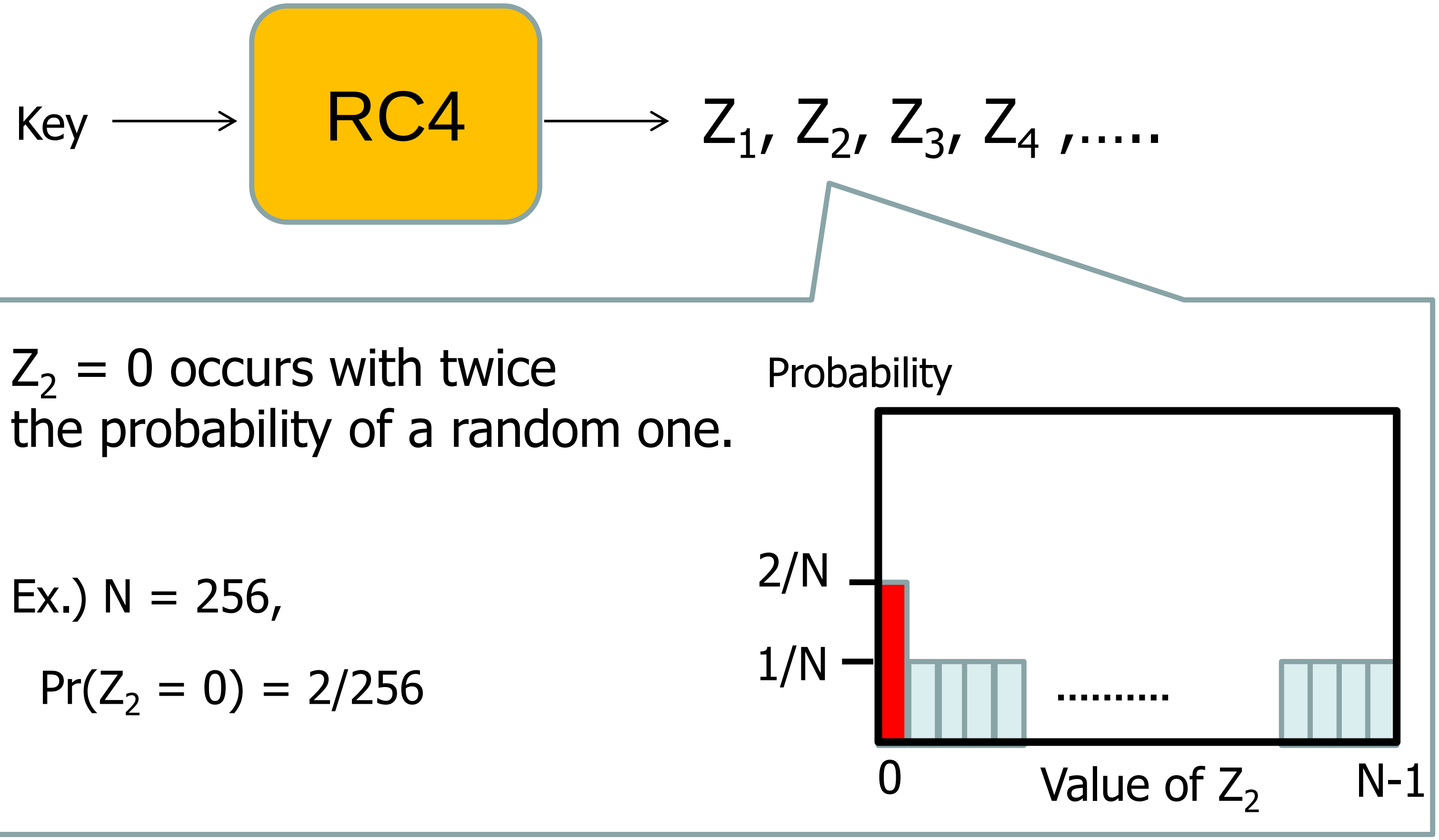
- Stream Cipher designed by Ron Rivest in 1987
  - is widely used, e.g. SSL/TLS, WEP/WPA and more.
- Parameter
  - 1-256 byte key (typically 16 byte (=128 bit) key)
  - State size N bytes (typically N = 256)

We focus on
- 16 byte (128 bit) key
- 256 byte state



Mantin-Shamir Attack [MS01]

- Proposed in FSE 2001
- Second byte of the keystream is strongly biased to “0”



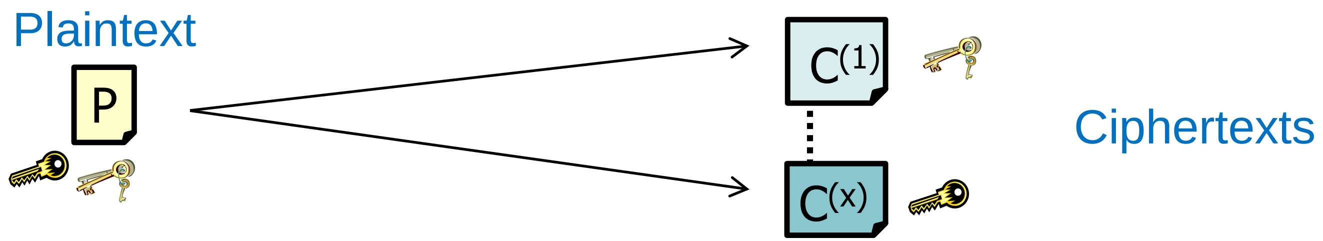
$Z_2 = 0$ occurs with twice the probability of a random one.

Ex.) $N = 256$,

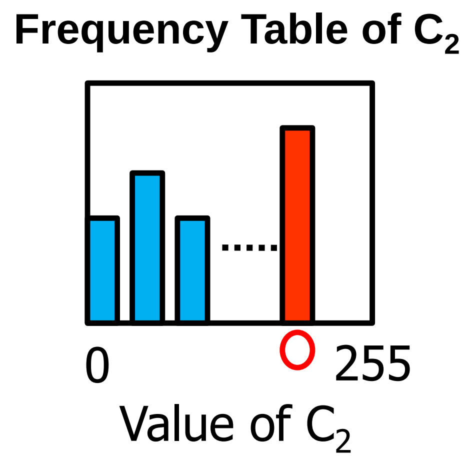
$Pr(Z_2 = 0) = 2/256$

Plaintext Recovery Attack [MS01]

- **Broadcast setting** : same plaintext is encrypted with different keys



- **Relation** : "$C_2 = P_2$ XOR Z_2"
  - If $Z_2 = 0$ (strong bias), then $C_2 = P_2$
  - Most frequent value of C_2 can be regarded as P_2



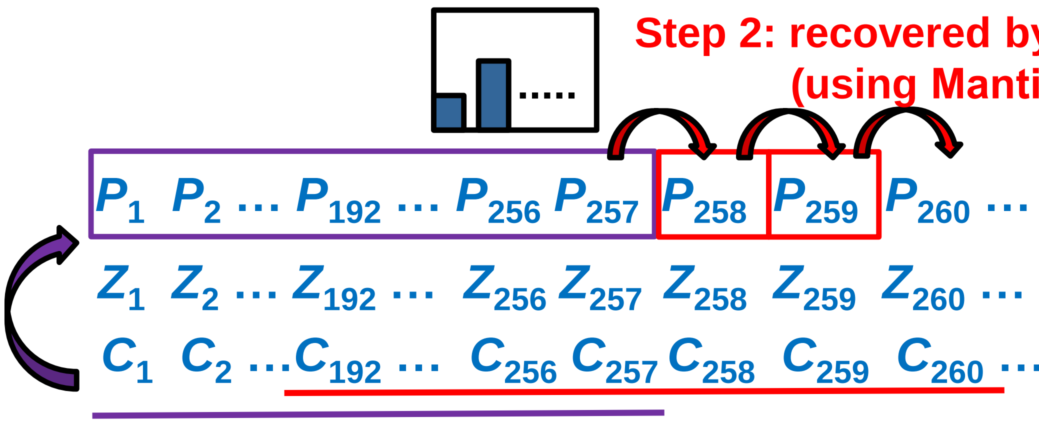
- **Evaluation**
  - Given Ω (N) ciphertexts encrypted by different keys,
    P_2 can be extracted with higher probability than a random search

Plaintext Recovery Attack in FSE 2013

HIROSHIMA UNIVERSITY

- Proposed by Isobe, Ohigashi, Watanabe and Morii
- is constructed by two phases
  - Initial byte recovery phase: recover initial 257 bytes of a plaintext
  - Sequential recovery phase: recover the later bytes of a plaintext using a knowledge of the first 257 bytes of a plaintext

Step 2: recovered by the sequential recovery phase (using Mantin's long-term bias)



Step 1: Recovered by the initial bytes recovery phase

Other previous attacks are also included

Conditional bias $Z_1=0|Z_2=0$

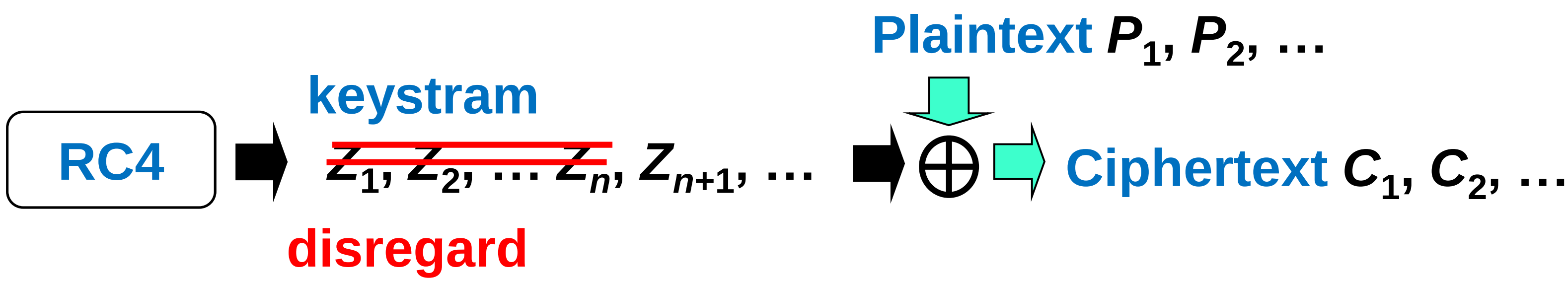
Single byte biases:

$Z_2 = 0$, $Z_3 = 131$, $Z_4 = 0$, $Z_r = r$ for $r = 5 \ldots 31$, $Z_0 = 0$ for $r = 32 \ldots 256$

$Z_r = -r$ for $r = 16,32,48,64,80,96,112$, $Z_{257} != 0$ (negative bias)

Countermeasure: RC4-drop

- is relatively secure RC4 implementation
- disregards the first n bytes of a keystream of RC4
  - recommendation(conservative) : n=3072



Initial byte biases are removed in RC4-drop

(Initial bytes recovery phase does not work)

Previous Attacks does not work on RC4-drop

Agenda

HIROSHIMA UNIVERSITY

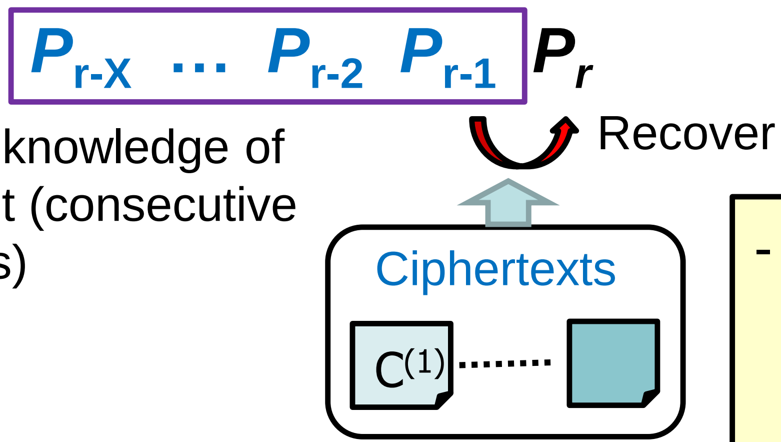
- RC4 Stream Cipher
- Previous Plaintext Recovery Attacks
- Plaintext Recovery Attack using Known Partial Plaintext Bytes
- Guess-and-Determine Plaintext Recovery Attack
- Conclusion

Plaintext Recovery Attack using Known Partial Plaintext Bytes

- is simply extension of FSE 2013 attack
  - **use partial knowledge of a target plaintext**
  - Based on **sequential recovery phase (Mantin's long-term bias)**

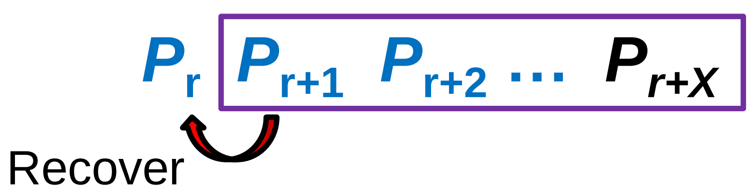
Forward attack function

Partial knowledge of a target (consecutive X bytes)



- The success probability increases with the increasing the value of X (when X < 67)
- If X=66, then the function is equivalent to that of sequential recovery phase of FSE 2013 attack

Backward attack function



Attack Procedure

- Example: consecutive 6 bytes of a target plaintext are known

Pre-known

$P_{r-6}\ P_{r-5} \cdots P_{r-2}\ P_{r-1}$ P_r — recover P_r with X = 6

$P_{r-6}\ P_{r-5} \cdots P_{r-2}\ P_{r-1}\ P_r$ P_{r+1} — recover P_{r+1} with X = 7

⋮

$P_{r-6}\ P_{r-5} \cdots P_{r-2}\ P_{r-1}\ P_r\ P_{r+1} \cdots P_{r+59}$ P_{r+60} — recover P_{r+60} with X = 66

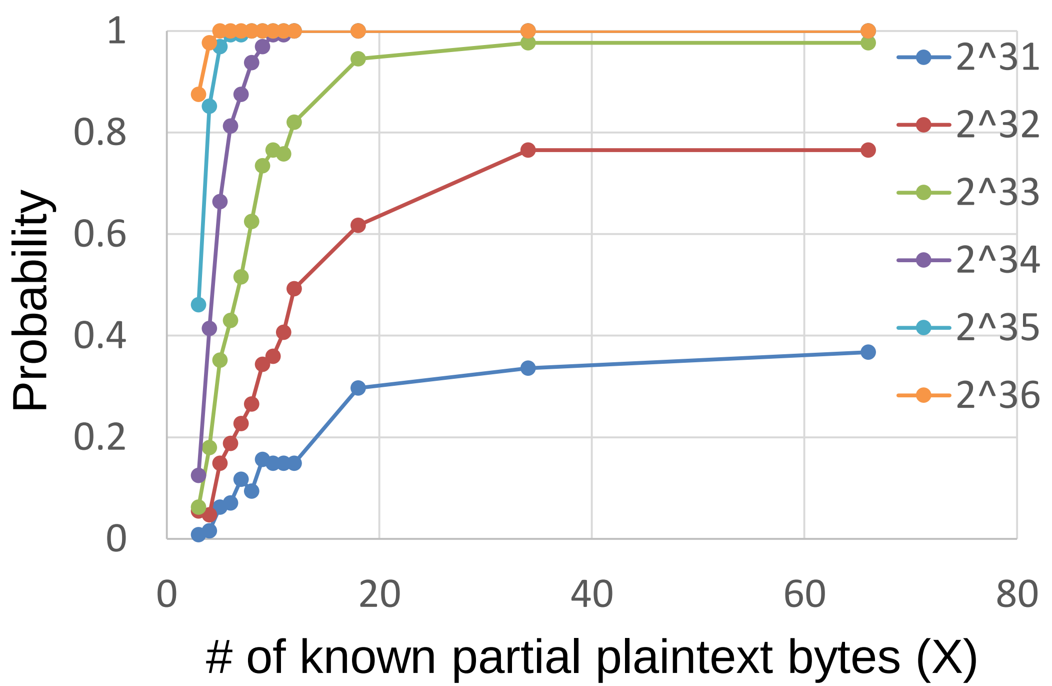
P_{r-6} $P_{r-5} \cdots P_{r-2}\ P_{r-1}\ P_r\ P_{r+1} \cdots P_{r+59}\ P_{r+60}$ P_{r+61} — recover P_{r+61} with X = 66

⋮ (later processes are similar to FSE2013 attack)

Experimental Result

HIROSHIMA UNIVERSITY

- Probability for recovering (X+1)th byte of a plaintext using the knowledge of X bytes of the plaintext on RC4-drop(3072)
- Obtained from 128 test
- # of ciphertexts:
  $2^{31}, 2^{32}..., 2^{36}$
- X = 3, 4, …, 66

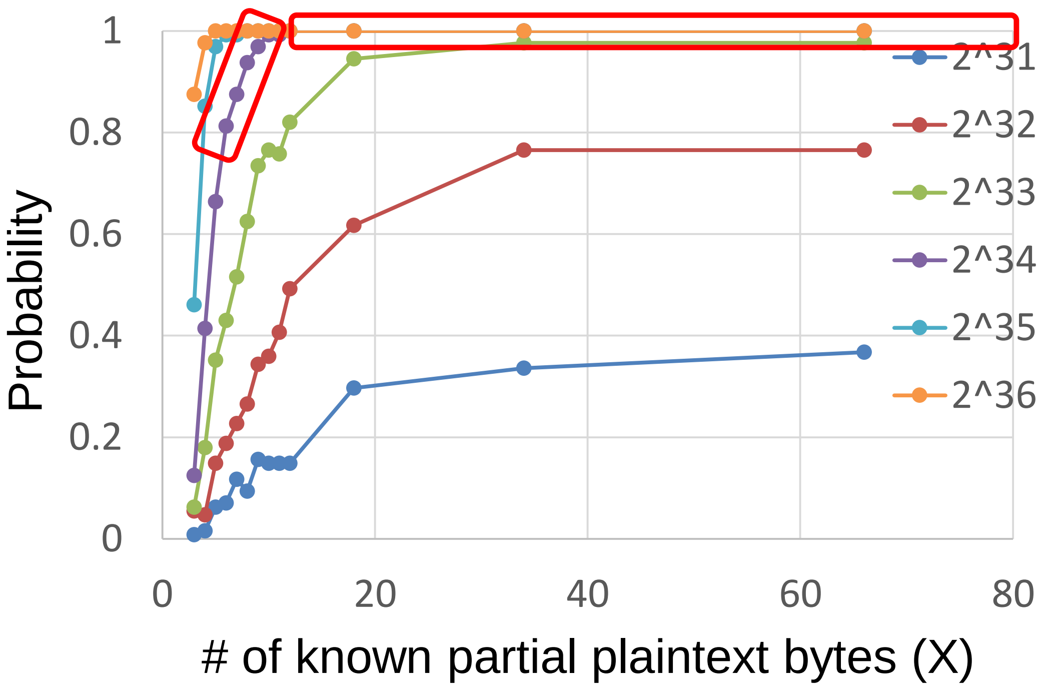


Evaluation

ex.) consecutive 6 bytes of a target plaintext and 2^{34} ciphertexts are given
Consecutive 1petabyte of plaintext are recovered with probability of

Experimental Result

- Probability for recovering (X+1)th byte of a plaintext using the knowledge of X bytes of the plaintext on RC4-drop(3072)
- Obtained from 128 test
- # of ciphertexts: $2^{31}, 2^{32}\ldots, 2^{36}$
- X = 3, 4, …, 66



Evaluation

ex.) consecutive 6 bytes of a target plaintext and 2^{34} ciphertexts are given

Consecutive 1petabyte of plaintext are recovered with probability of

$$\mathbf{0.8125 \times 0.8750 \times 0.9375 \times 0.9688 \times 0.9922 \times 0.9922 \sim 0.636}$$

Agenda

- RC4 Stream Cipher
- Previous Plaintext Recovery Attacks
- Plaintext Recovery Attack using Known Partial Plaintext Bytes
- Guess-and-Determine Plaintext Recovery Attack
- Conclusion

Guess and Determine Plaintext Recovery Attack

- does not require any previous knowledge of a plaintext
- uses attack functions based on two long-term biases
  - Mantin's long-term bias in EUROCRYPT 2005 (ABSAB bias)
  - Fluhrer-McGrew long-term bias in FSE 2000 (FM00 bias)

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

P_{r-X} … P_{r-2} P_{r-1} P_r

$f_{ABSAB\_F}()$ Recover

P_r P_{r+1} P_{r+2} … P_{r+X}

Recover $f_{ABSAB\_B}()$

Attack function based on FM00 bias (NEW)
(conditional bias of FM00 bias)

P_{r-1} P_r

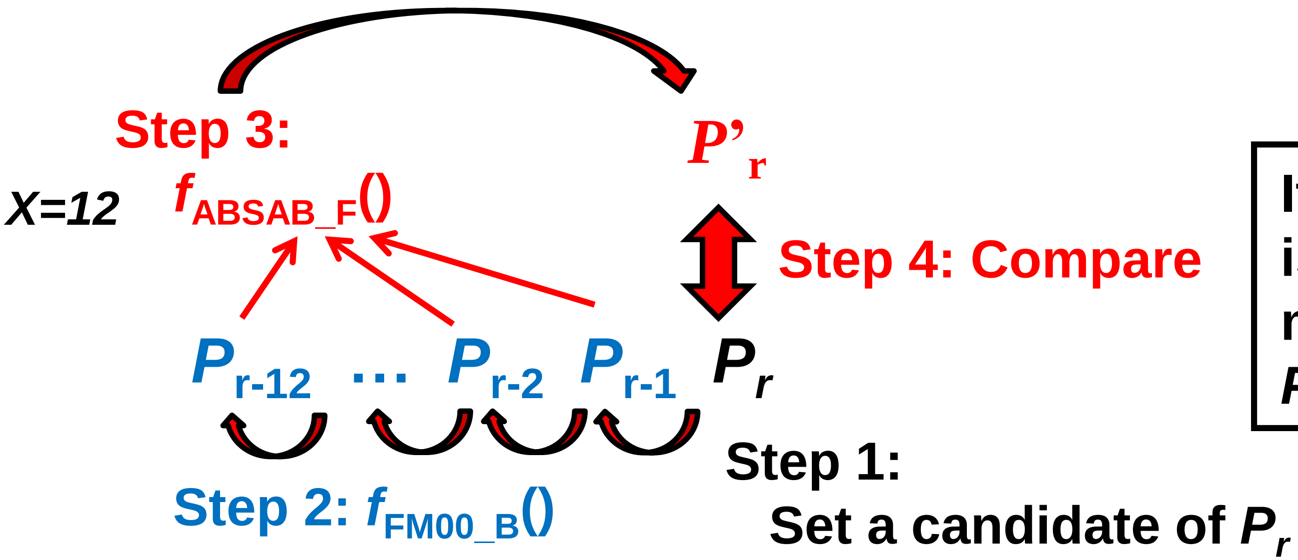
$f_{FM00\_F}()$ Recover

P_{r-1} P_r

Recover $f_{FM00\_B}()$

Attack Procedure

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



If # of candidates of P_r is not one, the same method is repeated for P'_{r-1}, P'_{r-2}, …

Experimental Result

- 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, our attack can recover any plaintext byte with probability close to one
- Given 2^{34} ciphertexts, our attack can recover any plaintext byte with probability of about 0.91

Conclusion

Security Evaluation of RC4-drop in the Broadcast/Multi-session Setting

Results

- **Plaintext recovery attack using Known Partial Plaintext Bytes**
  - Given consecutive 6 bytes of a target plaintext and 2^{34} ciphertexts with different keys, consecutive 1 petabytes of the plaintext are recovered with probability of more than 0.6

- **Guess-and-Determine Plaintext Recovery Attack**
  - Not Require any previous knowledge of a plaintext
  - Given $\mathbf{2^{35}}$ ciphertexts with different keys, any position of the plaintext byte is recovered with probability of close to one

RC4 is not secure even if initial keystream bytes are dropped