WEP Weak IVs Revisited

Kazukuni Kobara and Hideki Imai

IIS, Univ. of Tokyo

RCIS, AIST

Outline

- Available options for securing WLAN access
- WEP and its key recovery attack
- Condition to recover the WEP key
- Good and bad strategies to trace the condition back to the patterns of IVs and WEP keys
- Conclusion

Available Options for Securing WLAN Access

- Channel Protection (& Authentication)
  - AES-CCM
  - TKIP
  - (Weak-IV skipping WEP)
  - WEP
- Filtering
  - Filtering with MAC address
- (Authentication &) Key-Establishment
  - EAP-TLS
  - EAP-TTLS, PEAP
  - EAP-MD5, LEAP
  - PSK

Disadvantage:

•Old WLAN cards and APs cannot support them

Current Status

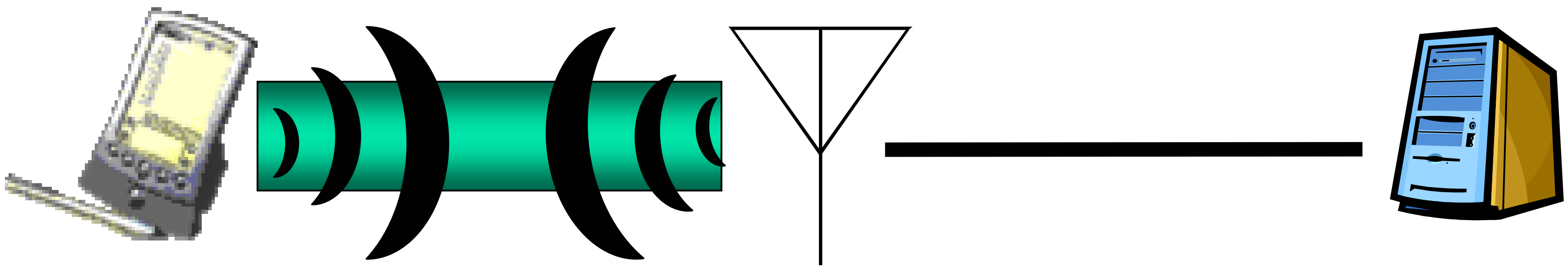
▪ AES-CCM ▪ TKIP	Fully investigated and no serious attack has been identified
▪ (Weak-IV skipping) WEP	Not fully investigated
▪ (Conventional) WEP ▪ Filtering with MAC address	Insecure even against casual attacks

Advantage:

•Compatible with WEP
•Old WLAN cards and APs may support easily

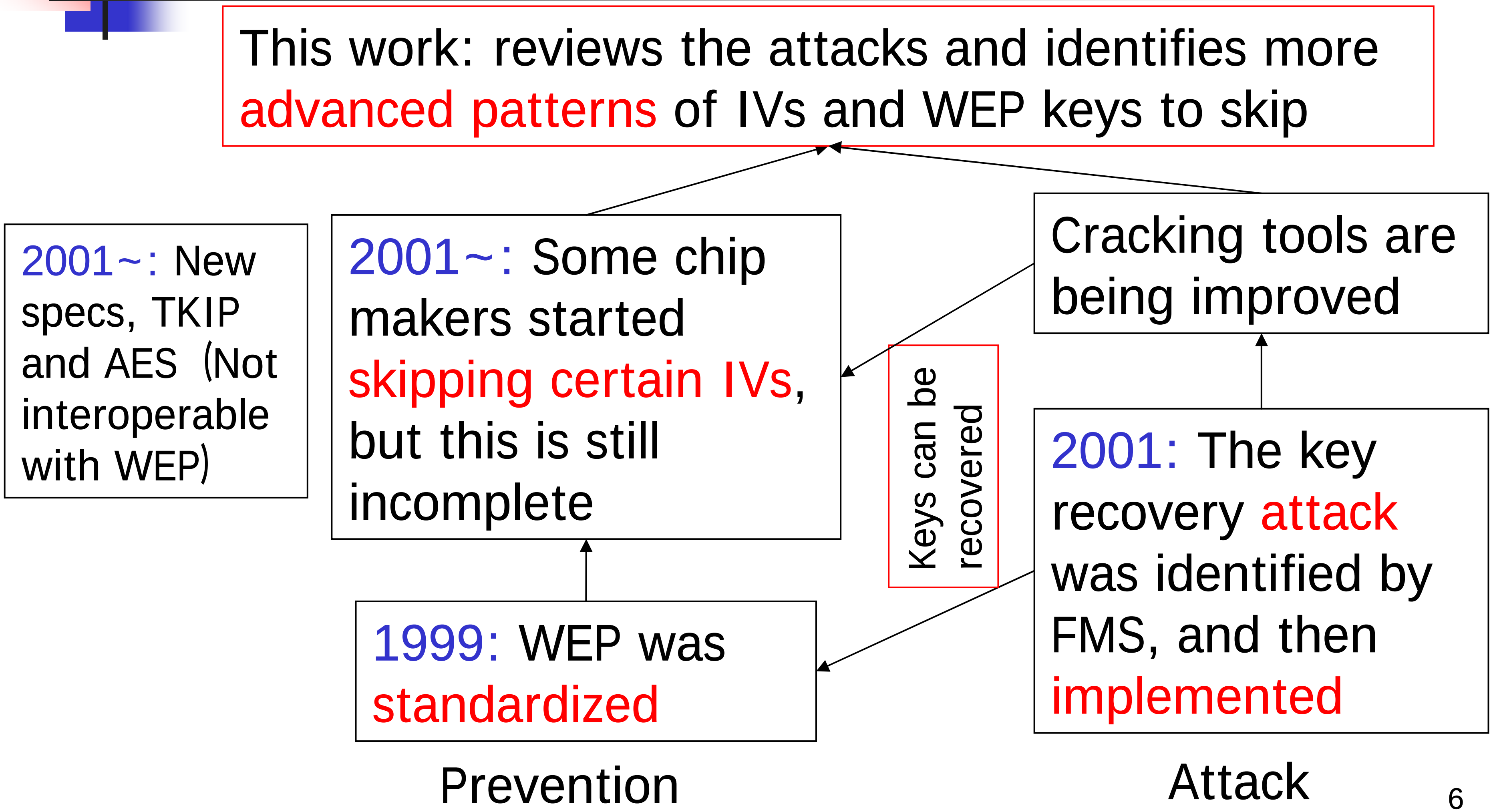
WEP: Wired Equivalent Privacy

- A specification for securing wireless access, especially of 802.11



Note: WEP (as well as TKIP and AES-CCM) give protection only for wireless part, but not for the wired part.

History of battles over WEP

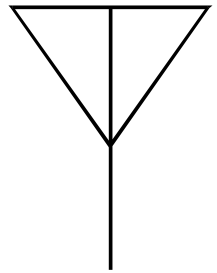


WEP：
Wired Equivalent Privacy

mobile node

access point

IV, (m||CRC(m))+RC4(IV||K')



Pre-Shared Key: K'

Pre-Shared Key: K'

IV: Initial Value

m: message

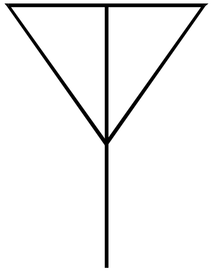
+: exclusive-or

||: concatenation

WEP : Wired Equivalent Privacy

mobile node

access point



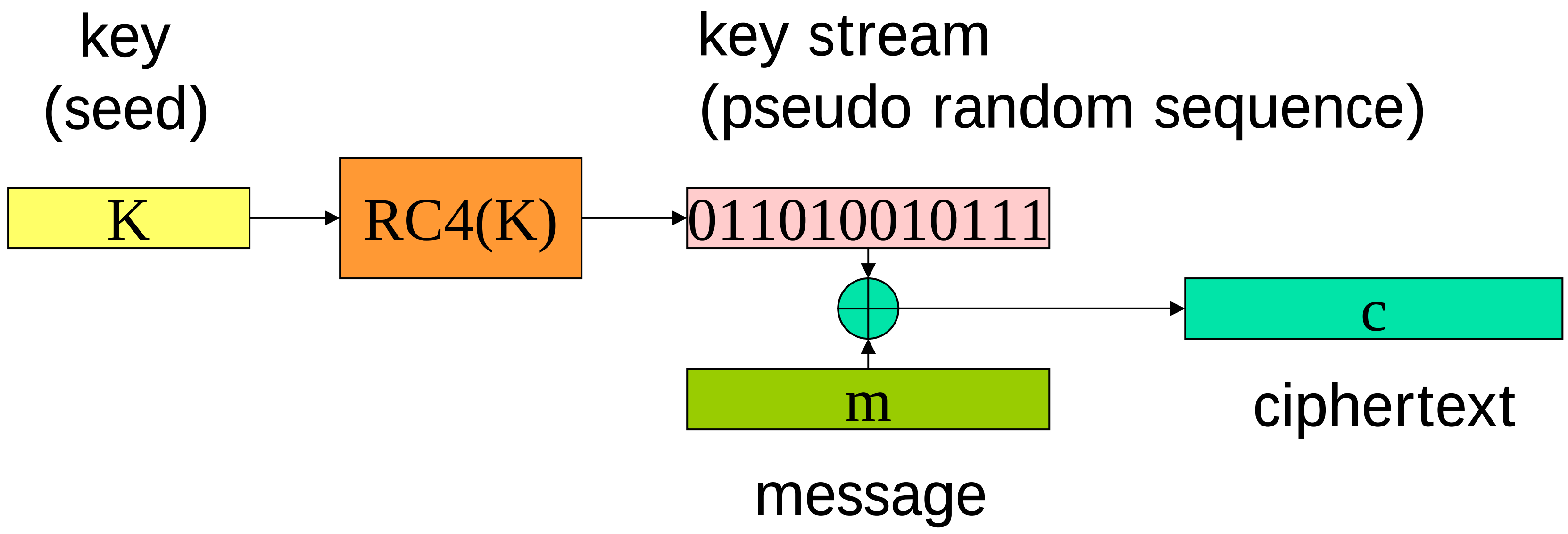
IV, (m||CRC(m))+RC4(IV||K')

Integrity check

Encryption
with RC4 key stream

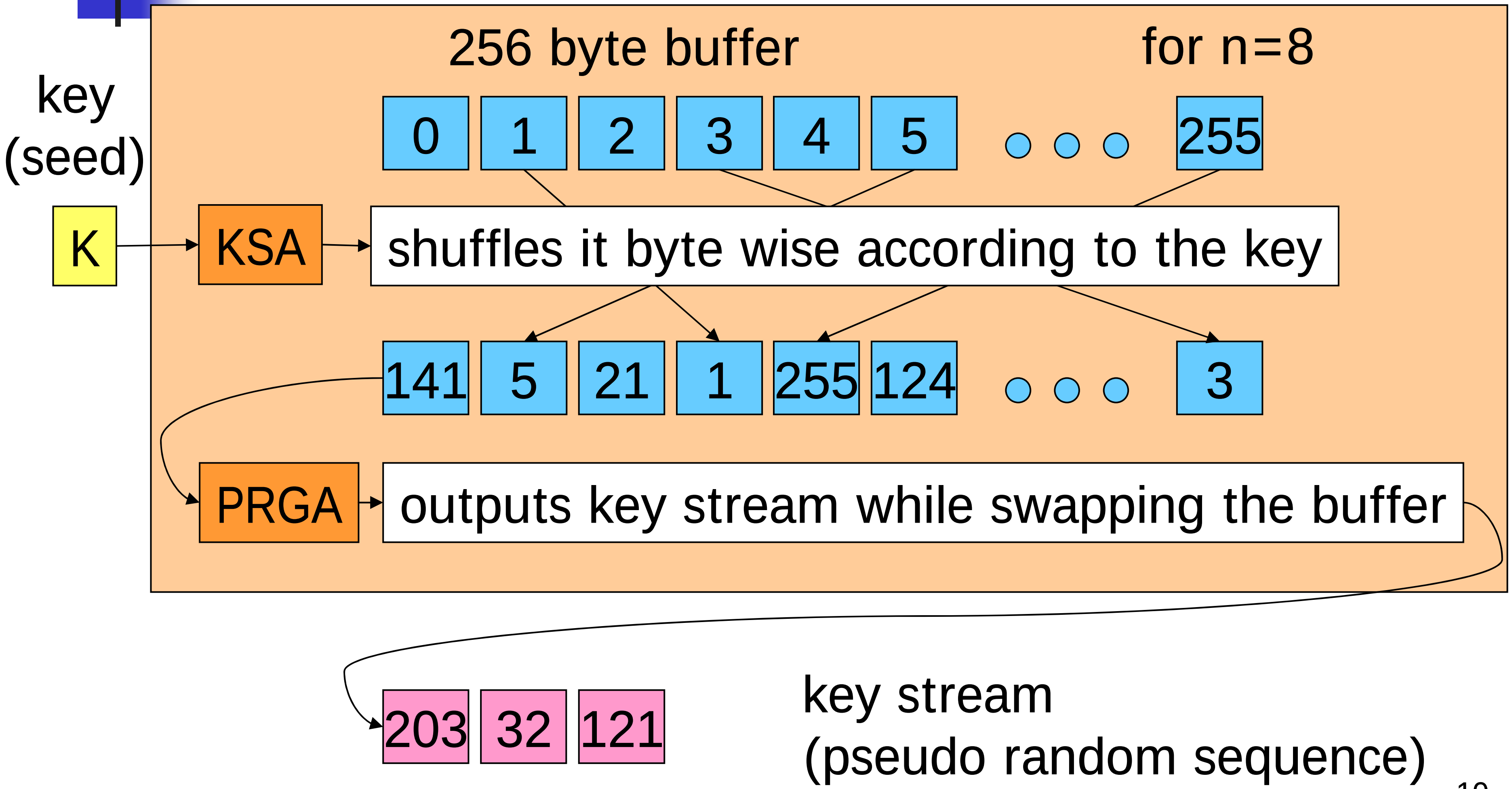
+: exclusive-or

RC4 Stream Cipher



KSA: Key Scheduling Algorithm
PRGA: Pseudo Random Generator Algorithm

RC4



KSA

Input: byte size n, a l byte key K

Output: a 2^n byte buffer S

$$
\begin{aligned}
& S := (0, 1, \cdots, 2^n - 1) \\
& j := 0 \\
& \text{For}(i = 0; i < 2^n; i++)\{ \\
& \qquad j := j + S[i] + K[i \mod l] \mod 2^n \\
& \qquad \text{Swap } S[i] \text{ and } S[j] \\
& \} \\
& \text{Return } S
\end{aligned}
$$

PRGA

Input: output sequence size s and a 2^n byte buffer S

Output: output sequence Z

$j := 0$

For$(i' = 1; i' \leq s; i' + +)\{$

$i := i' \mod 2^n$

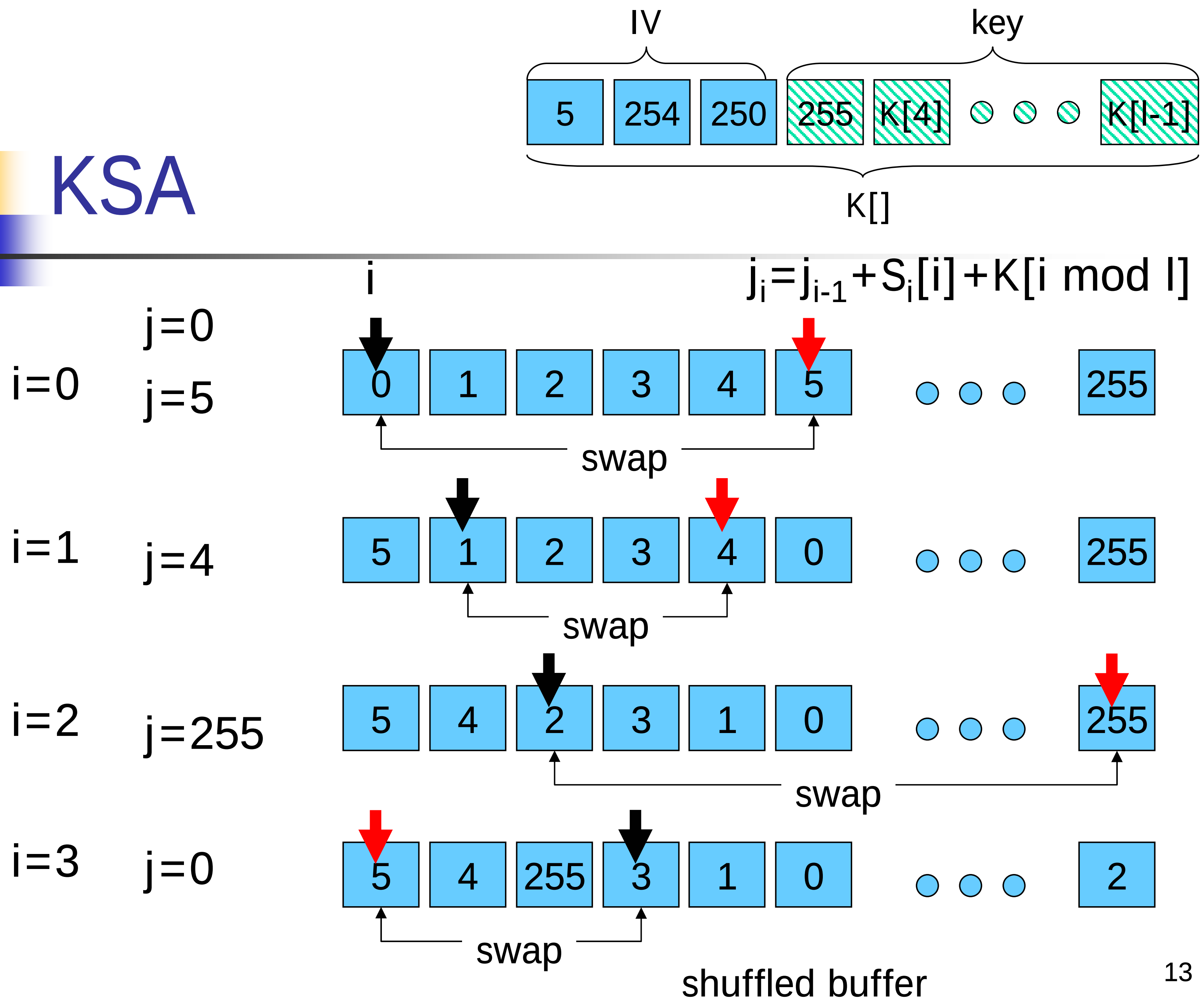
$j := j + S[i] \mod 2^n$

Swap $S[i]$ and $S[j]$

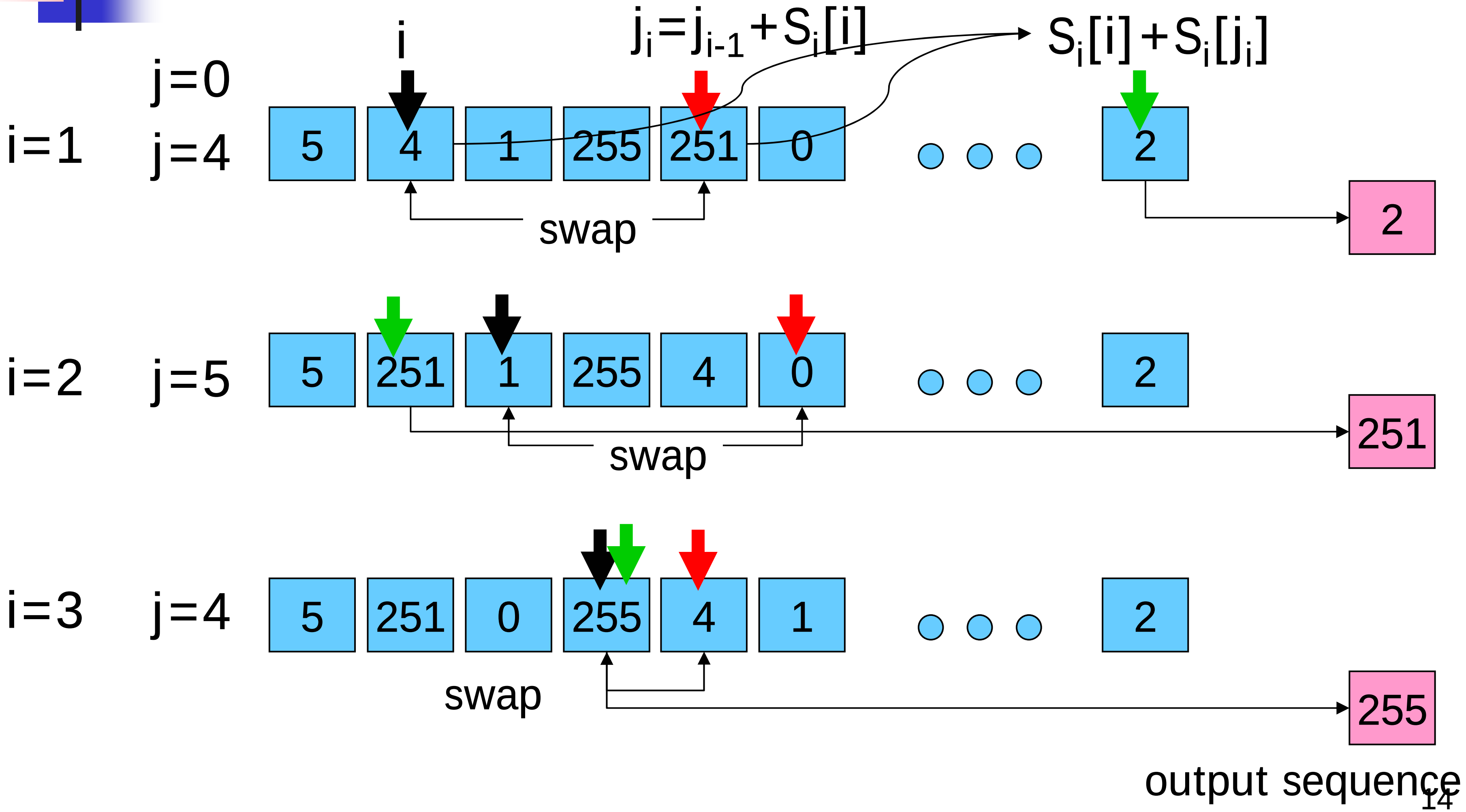
$Z[i' - 1] := S[S[i] + S[j] \mod 2^n]$

Return $Z[i' - 1]$ }

KSA


IV
key
5
254
250
255
K[4]
K[l-1]
K[]
i
$j_i=j_{i-1}+S_i[i]+K[i \bmod l]$
j=0
i=0
j=5
0 1 2 3 4 5 255
swap
i=1
j=4
5 1 2 3 4 0 255
swap
i=2
j=255
5 4 2 3 1 0 255
swap
i=3
j=0
5 4 255 3 1 0 2
swap
shuffled buffer

PRGA



i
$j_i = j_{i-1} + S_i[i]$
$S_i[i] + S_i[j_i]$
j=0
i=1
j=4
5
4
1
255
251
0
2
swap
2
i=2
j=5
5
251
1
255
4
0
2
swap
251
i=3
j=4
5
251
0
255
4
1
2
swap
255
output sequence

Gap between WEP and others

WEP

[FMS01][SIR01]

known — unknown

IV, RC4(IV || key) → Key is recoverable

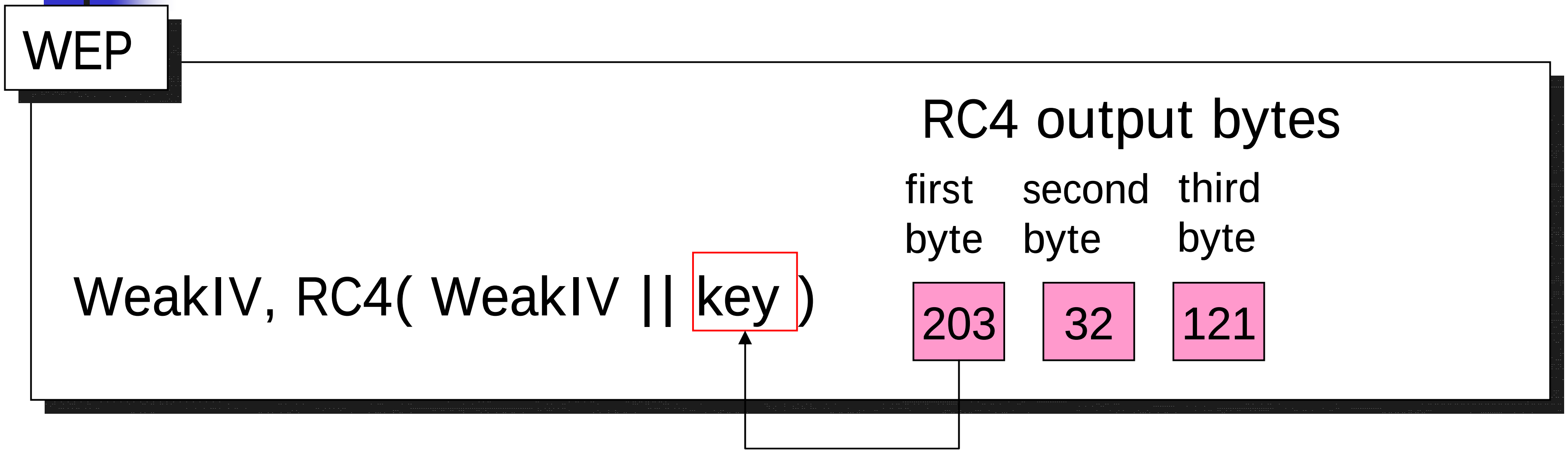
While the gap might be small, it made a big difference!!

SSL/TLS etc

unknown

RC4(key) → key is not recoverable

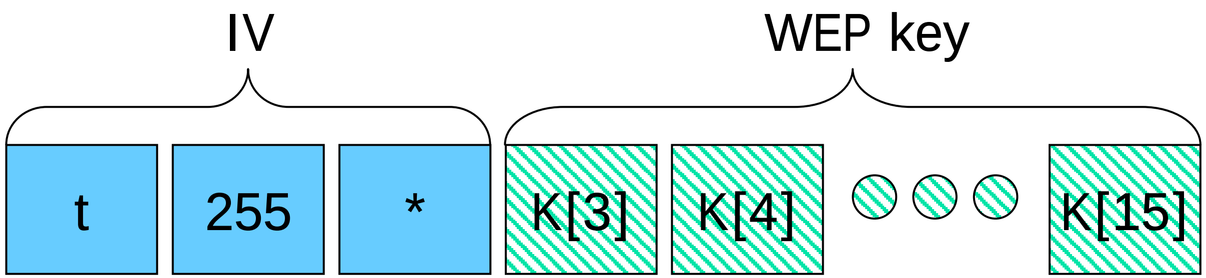
Idea of Key Recovery Attack



For certain IVs called “Weak IVs” the correlation between the first output byte and one byte of the key becomes higher than the average 1/256=0.004.

Typical prob. is 0.05

The famous weak IVs identified by FMS



t=3 to 15

t: target key byte to crack

Notations

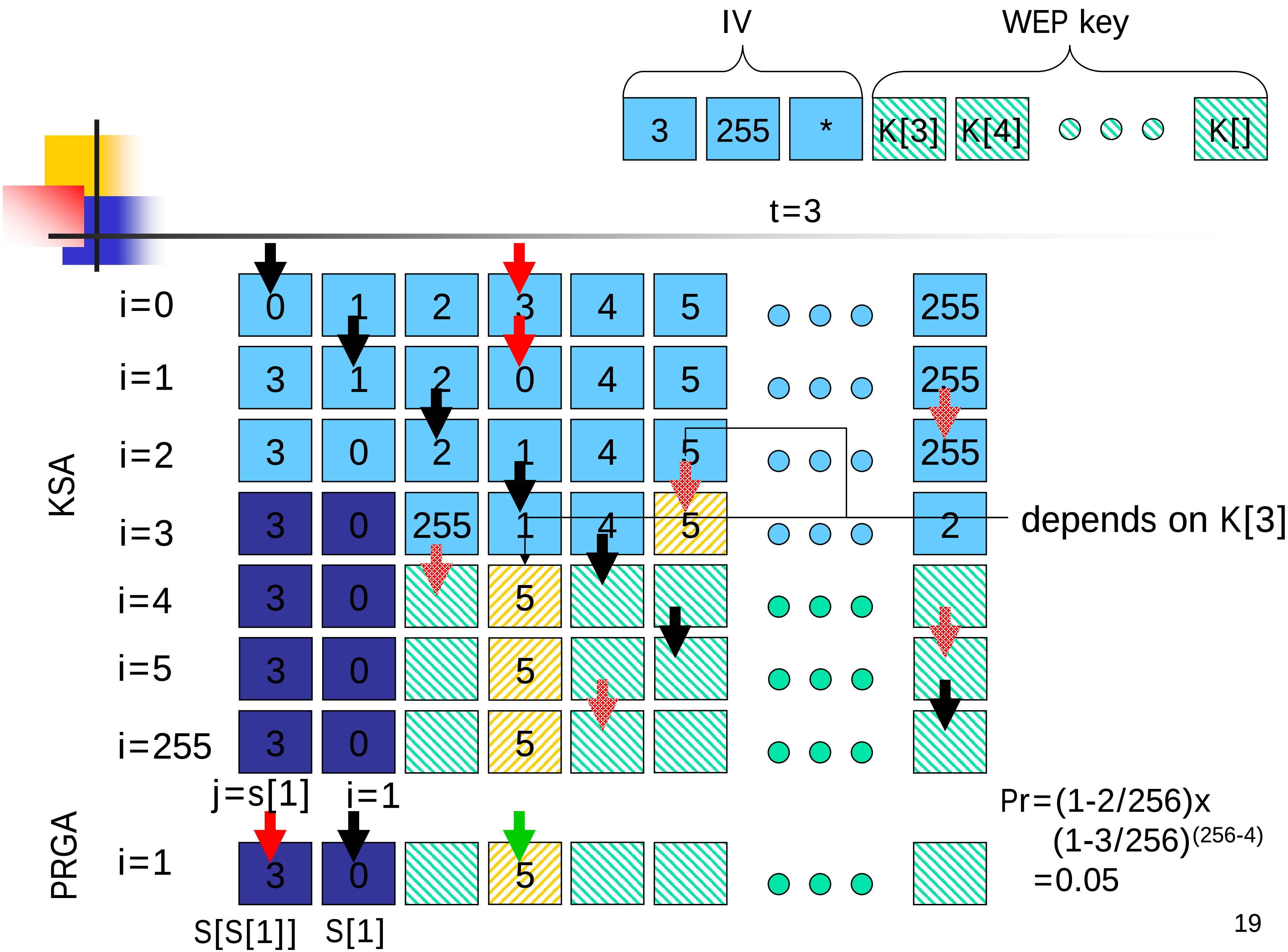
Known byte

Known and untouchable byte
(should not be referred to by index j_i for $i > t'$)

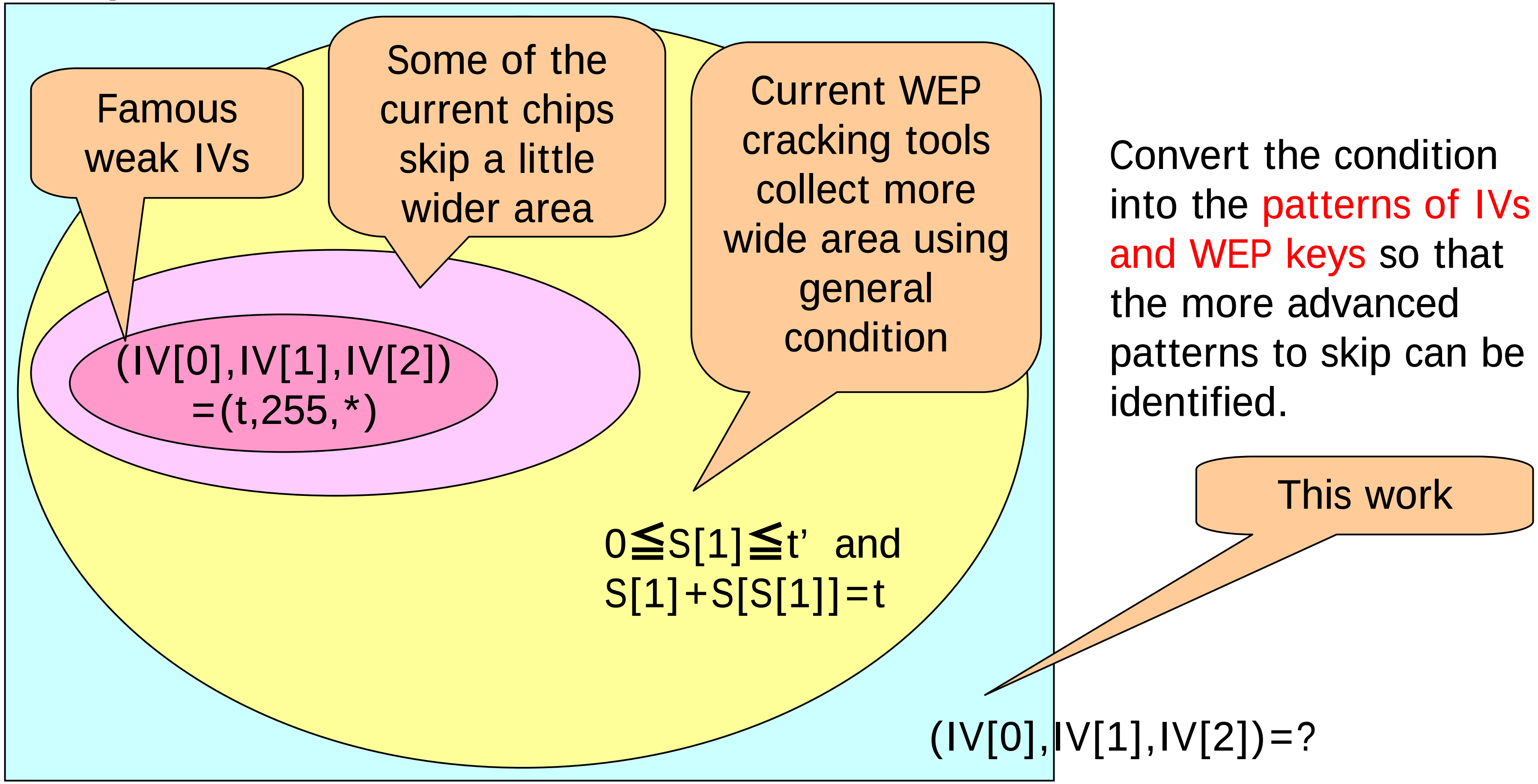
Target byte
(which depends on K[t] and
should not be referred to by j_i for $i > t'$ except $i=t$)

Unknown byte

t' : (# of known bytes in K[])-1


IV
WEP key
3
255
*
K[3]
K[4]
K[]
t=3
KSA
i=0
0
1
2
3
4
5
255
i=1
3
1
2
0
4
5
255
i=2
3
0
2
1
4
5
255
i=3
3
0
255
1
4
5
2
depends on K[3]
i=4
3
0
5
i=5
3
0
5
i=255
3
0
5
j=s[1]
i=1
PRGA
i=1
3
0
5
S[S[1]]
S[1]
Pr=(1-2/256)x
(1-3/256)^(256-4)
=0.05

Relationship Among Weak IVs



Convert the condition into the patterns of IVs and WEP keys so that the more advanced patterns to skip can be identified.

Note (K[0], K[1], K[2])=(IV[0], IV[1], IV[2])

The difficult part

- S[] depends not only on IVs, but also on WEP keys, K[3] to K[t']
  - i.e. by exhaustive searching K[3] to K[t'], a lot of key-dependent weak IVs are available
  - (and skipping key-dependent weak IVs only is not enough!!)
- Listing up all the combinations of IVs and WEP keys with exhaustive search is computationally infeasible

Another Naive Approach

- Skip IVs meeting the condition but only for the currently set WEP key
  - This is feasible, but
- This causes another vulnerability
  - the information on the WEP key is revealed from the skipped patterns
  - since most of the weak IVs depend on the WEP key

We took the approach

- to trace the condition back to the patterns of IVs and WEP keys theoretically

- We are now summarizing the results and will open them soon

Our Contribution