Applied Cryptography
CPEG 472/672
Lecture 5A

Instructor: Nektarios Tsoutsos

Software-oriented SCs

- Interest in software SCs
  - Cheap HW and fast CPUs
  - Popularity compared to block ciphers after padding oracle attacks in CBC
  - Easier to specify compared to block ciphers
- Two interesting paradigms
  - RC4 (Rivest Cipher 4 – 1987)
    - Reverse engineered – leaked (1994)
    - Widely used for a long time (e.g., WEP, TLS)
  - Salsa20

How RC4 works

- Simply swap bytes in state array
  - No S-boxes, XOR or nonlinear ops, no mul
- Initialize state array (key K is *n* bytes):

```
j = 0 #index
S = range(256)
for i in range(256):
    j = (j + S[i] + K[i % n]) % 256
    S[i], S[j] = S[j], S[i]
```

Initialized state array: Permutation of byte values 0 to 255

RC4 keystream generation

Took 20 years to find exploitable flaws

- Generate keystream from initial state

```
i = 0    #index
j = 0    #index
for b in range(m):
    i = (i + 1) % 256
    j = (j + S[i]) % 256
    S[i], S[j] = S[j], S[i]  #swap
    KS[b] = S[(S[i] + S[j]) % 256]
```

S = state
m = message

RC4 failures: WEP

- WEP uses RC4 to encrypt 802.11 frames
  - All payloads in the same session use same key + a 3 byte public nonce in the header
- RC4 does not support a nonce
  - WEP designers did a hack
  - Prepend 3 nonce bytes to the key

What is the problem?

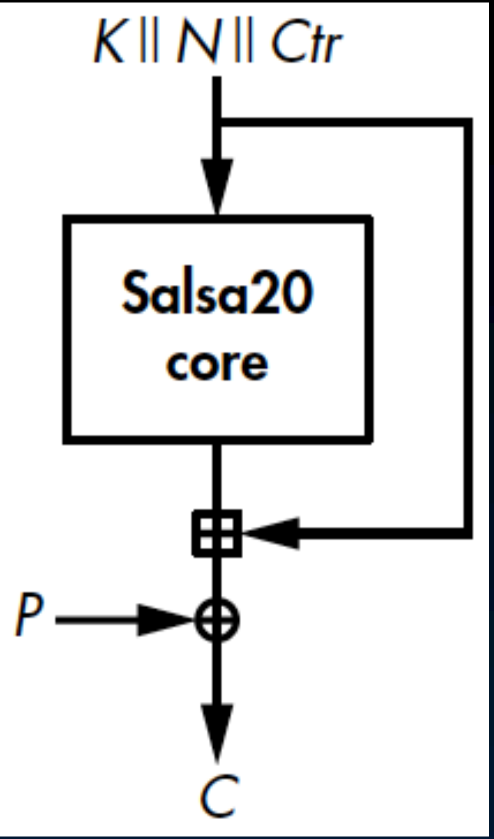
- Problems: nonce too small (can repeat), attacker can determine state on 3rd iter
  - 1st keystream byte depends on 1st key byte
  - Exploit: Known/chosen plaintext attack

RC4 failures: TLS

- Uses unique 128-bit key on RC4
  - Non public nonce mistake like WEP
- RC4 has statistical biases
  - 2^{nd} keystream byte == 0 with Prob 1/128
  - The probability should be 1/256 instead
  - All of first 256 bytes are biased as well
- Attack model: broadcast model
  - Need the same ptxt encrypted with different keys many times
  - Bias: KS bytes more likely to be zero

Salsa20

https://cr.yp.to/snuffle/spec.pdf



- CTR-based stream cipher
  - Salsa20 core – 512 bit blocks
  - Add input block to output
    - Otherwise the cipher is insecure
  - State: 4 x 4 32-bit words
- Salsa20 core
  - Quarter-Round (QR) permutation
    - 1 column round = 4 QR for 4 columns
    - 1 row round = 4 QR for 4 rows
  - 1 double round = 1 column + 1 row round
    - 10 double rounds total

Salsa20

- Constant c, key k, nonce v, ctr t

c_0	k_0	k_1	k_2
k_3	c_1	v_0	v_1
t_0	t_1	c_2	k_4
k_5	k_6	k_7	c_3

- Nothing up my sleeves constants
  - "expand 32-byte k" broken in 4 parts

Salsa20

https://cr.yp.to/snuffle/spec.pdf

- z0, z1, z2, z3 = **QR** (y0, y1, y2, y3)

$$
\begin{aligned}
z_1 &= y_1 \oplus ((y_0 + y_3) \lll 7), \\
z_2 &= y_2 \oplus ((z_1 + y_0) \lll 9), \\
z_3 &= y_3 \oplus ((z_2 + z_1) \lll 13), \\
z_0 &= y_0 \oplus ((z_3 + z_2) \lll 18).
\end{aligned}
$$

Start with z1

- y1 changes to z1, then y2 changes to z2
- Then y3 changes to z3
- Finally, y0 changes to z0 (order matters)

Salsa20

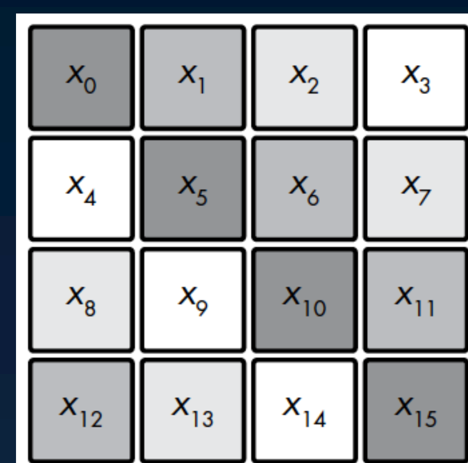
https://cr.yp.to/snuffle/spec.pdf

- Column round

$$
\begin{aligned}
(y_0, y_4, y_8, y_{12}) &= \mathrm{quarterround}(x_0, x_4, x_8, x_{12}),\\
(y_5, y_9, y_{13}, y_1) &= \mathrm{quarterround}(x_5, x_9, x_{13}, x_1),\\
(y_{10}, y_{14}, y_2, y_6) &= \mathrm{quarterround}(x_{10}, x_{14}, x_2, x_6),\\
(y_{15}, y_3, y_7, y_{11}) &= \mathrm{quarterround}(x_{15}, x_3, x_7, x_{11}).
\end{aligned}
$$

- State

$$
\begin{pmatrix}
x_0 & x_1 & x_2 & x_3\\
x_4 & x_5 & x_6 & x_7\\
x_8 & x_9 & x_{10} & x_{11}\\
x_{12} & x_{13} & x_{14} & x_{15}
\end{pmatrix}
$$



Salsa20

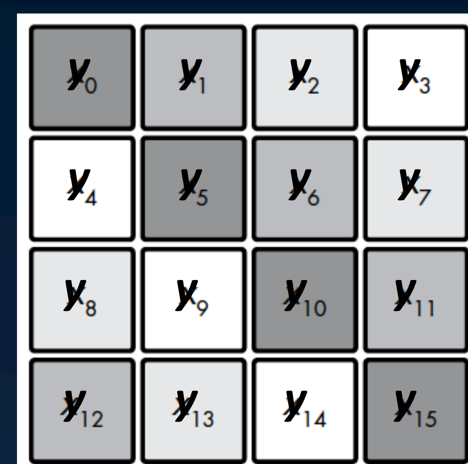
https://cr.yp.to/snuffle/spec.pdf

- Row round

$$
\begin{aligned}
(z_0, z_1, z_2, z_3) &= \text{quarterround}(y_0, y_1, y_2, y_3),\\
(z_5, z_6, z_7, z_4) &= \text{quarterround}(y_5, y_6, y_7, y_4),\\
(z_{10}, z_{11}, z_8, z_9) &= \text{quarterround}(y_{10}, y_{11}, y_8, y_9),\\
(z_{15}, z_{12}, z_{13}, z_{14}) &= \text{quarterround}(y_{15}, y_{12}, y_{13}, y_{14}).
\end{aligned}
$$

- State

$$
\begin{pmatrix}
y_0 & y_1 & y_2 & y_3\\
y_4 & y_5 & y_6 & y_7\\
y_8 & y_9 & y_{10} & y_{11}\\
y_{12} & y_{13} & y_{14} & y_{15}
\end{pmatrix}
$$



Evaluation of Salsa20

- Two initial states – 1 bit difference

61707865	00000000	00000000	00000000	61707865	00000000	00000000	00000000
00000000	3320646e	ffffffff	ffffffff	00000000	3320646e	ffffffff	ffffffff
00000000	00000000	79622d32	00000000	**00000001**	00000000	79622d32	00000000
00000000	00000000	00000000	6b206574	00000000	00000000	00000000	6b206574

- After 10 double rounds

e98680bc	f730ba7a	38663ce0	5f376d93	1ba4d492	c14270c3	9fb05306	ff808c64
85683b75	a56ca873	26501592	64144b6d	b49a4100	f5d8fbbd	614234a0	e20663d1
6dcb46fd	58178f93	8cf54cfe	cfdc27d7	12e1e116	6a61bc8f	86f01bcb	2efead4a
68bbe09e	17b403a1	38aa1f27	54323fe0	77775a13	d17b99d5	eb773f5b	2c3a5e7d

Differential cryptanalysis

- XOR between 2 states over many rounds

00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000001	00000000	00000000	00000000
00000000	00000000	00000000	00000000

Differential cryptanalysis

- XOR between 2 states over many rounds

80040003	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000001	00000000	00000000	00000000
00002000	00000000	00000000	00000000

Differential cryptanalysis

- XOR between 2 states over many rounds

9ed7eb7f	060002c0	18028b0c	57ca83c0
00000000	00000000	00000000	00000000
00000001	0000e000	801c0006	00000000
00002000	00400000	04000008	0060f300

Differential cryptanalysis

- XOR between 2 states over many rounds

3ab3c25d	9f40a5c9	10070e30	07bd03c0
db1ee2ce	43ee9401	21a702c3	48fd800c
403c1e72	00034003	4dc843be	700b8857
5625b75b	09c00e00	06000348	23f712d4

Differential cryptanalysis

- XOR between 2 states over many rounds

```
d93bed6d a267bf47 760c2f9f 4a41d54b
0e03d792 7340e010 119e6a00 e90186af
7fa9617e b6aca0d7 4f6e9a4a 564b34fd
98be796d 64908d32 4897f7ca a684a2df
```

Stream cipher failures

- Nonce reuse
  - Devastating mistake
  - Identical keystreams -> two time pad attack
  - Word/Excel: same nonce when save doc
- Broken RC4
  - Optimized: Swap bytes using XOR
    - X = X xor Y
    - Y = X xor Y
    - X = X xor Y

What is the problem?

- Insecure SCs in satphones (GMR 1-2)

Reading for next lecture

- Aumasson: Chapter 6
- Check errata:
- https://nostarch.com/seriouscrypto#updates