

Cryptography CS 555



Lecture 6: Stream Ciphers (RC4 and LFSR) and Their Use in WEP and CSS

Department of Computer Sciences
Purdue University

Lecture Outline

- Stream cipher
- RC4
- WEP (in)security
- LFSR
- CSS (in)security



Review of One-Time Pad (OTP)

- To encrypt a plaintext of n bits (letters), a key is a **random** n -bit (or n -letter) string
- Has perfect secrecy
- The same key should never be used twice
- Key length is the same as plaintext

Stream Ciphers

- Idea: replace “random” in OTP by “pseudo rand”
- Use Pseudo Random Number Generator
- PRNG: $\{0,1\}^s \mapsto \{0,1\}^n$
 - expand a short (e.g., 128-bit) random seed into a long (e.g., 10^6 bit) string that “looks random”
- Secret key is the seed
- $E_{\text{seed}}[M] = M \oplus \text{PRNG}(\text{seed})$

Properties of Stream Ciphers

- Does not have perfect secrecy
 - security depends on properties of PRNG
- PRNG must be “unpredictable”
 - given consecutive sequence of bits output (but not seed), next bit must be hard to predict
- Typical stream ciphers are very fast
- Used in many places, often incorrectly
 - SSL(RC4), DVD (LFSR), WEP (RC4), etc.

Fundamental Weaknesses of Stream Ciphers

- If the same stream is used twice ever, then easy to break.
 - in particular, susceptible to known-plaintext attack
- Highly malleable
 - easy to change ciphertext so that plaintext changes are predictable, e.g., flip bits
- Weaknesses exist even if the PRNG is strong

The RC4 Cipher: Generation

- The cipher internal state consists of
 - a 256-byte array S , which contains a permutation of 0 to 255
 - total number of possible states is $256! \approx 2^{1700}$
 - two indexes: i, j

$i = j = 0$

Loop

$i = (i + 1) \pmod{256}$

$j = (j + S[i]) \pmod{256}$

$\text{swap}(S[i], S[j])$

$\text{output } (S[i] + S[j]) \pmod{256}$

End Loop

The RC4 Cipher: Initialization

- Generate the initial permutation from a key k ; maximum key length is 2048 bits
- First divide k into L bytes
- Then

```
for i = 0 to 255 do
    S[i] = i
j = 0
for i = 0 to 255 do
    j = (j + S[i] + k[i mod L]) (mod 256)
    swap (S[i], S[j])
```

RC4 Cryptanalysis

- Attack objectives against stream ciphers (PRNG)
 - recover seed from an output sequence
 - predict the next bit from past output
 - distinguish output from a PRNG and output from a truly random string
- No practical attack in the generation part
 - given a random initial state, best known algorithm takes time $O(2^{700})$ to recover the state

Weakness of RC4 Initialization

- The second output word of RC4 is zero with probability $1/128$ rather than $1/256$
- The first two output words are both zero with a probability $3/256^2$
- The first byte generated by RC4 leaks information about individual key bytes.
 - best to drop first 256 bytes of output



WEP (in)security

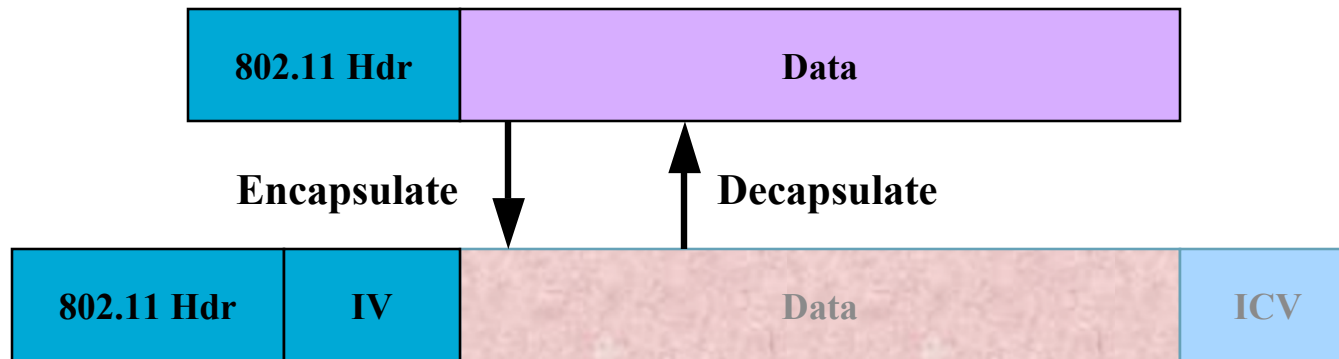


N. Borisov, I. Goldberg, D.
Wagner. [Intercepting Mobile
Communications: The Insecurity
of 802.11](#), MOBICOM 2001.

802.11 Security

- Used between a Wireless Access Point and Wireless Ethernet Cards
- Existing security consists of two subsystems
 - A data encapsulation technique called **W**ired **E**quivalent **P**rivacy (WEP)
 - An authentication algorithm called Shared Key Authentication
- Goals of 802.11 security
 - Data confidentiality
 - Access control
 - Data integrity

WEP Encapsulation



WEP Encapsulation Summary:

- A master key shared between the end points
- Encryption Algorithm = RC4
- Per-packet encryption key = 24-bit IV concatenated to a master key
- WEP allows IV to be reused
- Data integrity provided by CRC-32 of the plaintext data (the “ICV”)
- Data and ICV are encrypted under the per-packet encryption key

What Went Wrong in WEP?

- The space of IV is too small & IV is sent in clear
- With two messages are encrypted using the same IV, one can recover the key stream.
- The attack is made much easier by chosen plaintext attacks, which can be carried out in the environment where WEP is used.

Ways to Accelerate the Attack

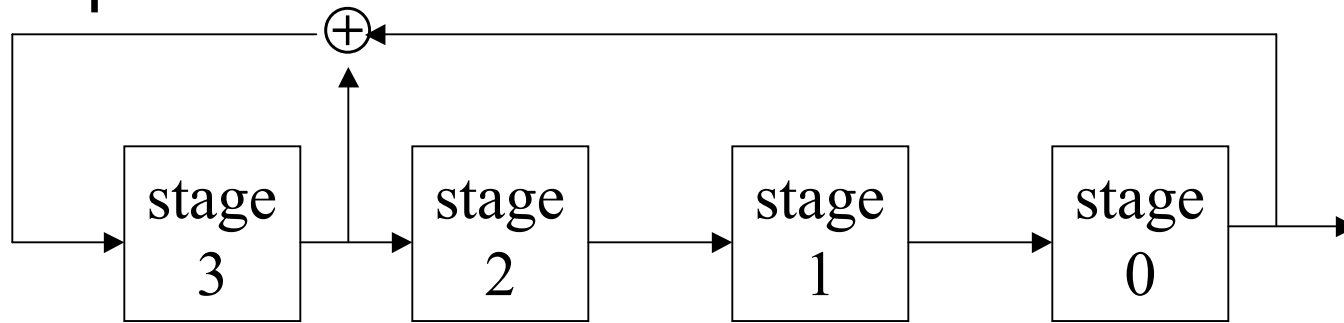
- Send spam into the network: no pattern recognition required!
- Get the victim to send e-mail to you
 - The AP creates the plaintext for you!
- Decrypt packets from one Station to another via an Access Point
 - If you know the plaintext on one leg of the journey, you can recover the key stream immediately on the other
- Etc., etc., etc.

Insecurity of shared key authentication

- The authentication scheme
 - send a random string (in clear) as a challenge
 - expects an corrected encrypted frame as response (encrypted using RC4)
- Anything wrong with this?

Linear Feedback Shift Register (LFSR)

- Example:



- $$z_i = z_{i-4} + z_{i-1} \pmod{2}$$
$$= 1 \cdot z_{i-1} + 0 \cdot z_{i-2} + 0 \cdot z_{i-3} + 1 \cdot z_{i-4} \pmod{2}$$
- Connection polynomial
 - $C(D) = 1 + 1 \cdot D + 0 \cdot D^2 + 0 \cdot D^3 + 1 \cdot D^4$

Maximum-length LFSR

- **Definition:** For a L -stage maximum-length LFSR, any non-zero initial state produces an output sequence with period equal to $2^L - 1$, this is called a m-sequence.
- **Fact:** The distribution of patterns having fixed length is almost uniform in a m-sequence.

Cryptanalysis of LFSR

- Given a 4-stage LFSR, we know
 - $z_4 = z_3c_3 + z_2c_2 + z_1c_1 + z_0c_0 \pmod 2$
 - $z_5 = z_4c_3 + z_3c_2 + z_2c_1 + z_1c_0 \pmod 2$
 - $z_6 = z_5c_3 + z_4c_2 + z_3c_1 + z_2c_0 \pmod 2$
 - $z_7 = z_6c_3 + z_5c_2 + z_4c_1 + z_3c_0 \pmod 2$
- Knowing z_0, z_1, \dots, z_7 , one can compute c_0, c_1, c_2, c_4 .

Usage of LFSR

- Easy to implement in hardware
- Multiple LFSR's are often combined to achieve better security

Content Scrambling System (CSS)

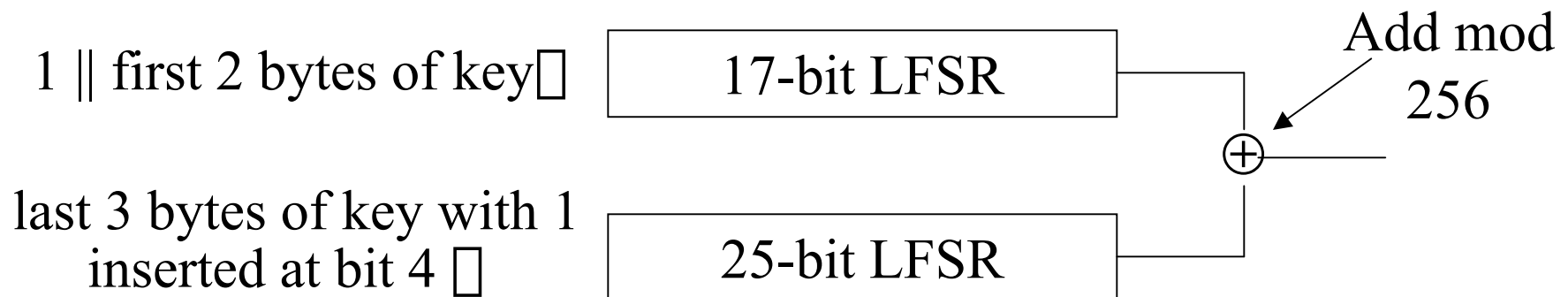
- Designed by Matsushita and Toshiba, and used for encrypting DVD videos
- There is a set of 409 player keys
- Each DVD player has one player key
- Each disk has a key data block
 - the disk key encrypted under the disk key (hash)
 - disk key encrypted with player key 1
 - ...
 - disk key encrypted with player key 409
- The disk key is used to encrypt title keys

Attacking CSS

- Knowing a disk key, by attacking the CSS cipher, one can recover all player keys
 - takes about 2^{25} time
 - breaks the revocation model of CSS
- It is possible to attack the hash to recover the disk key
 - takes about 2^{25} time

CSS Stream Cipher

- Key = 5 bytes = 40 bits
 - brute-force attack is possible
 - more efficient attacks exist



Given 6 output bytes, a trivial 2^{16} attack exists

A similar attack with 5 output bytes exists

Summary

- Stream cipher approximate OTP with PRNG
- Weakness
 - If the same stream is used twice ever, then easy to break.
 - Highly malleable
 - Weaknesses exist even if the PRNG is strong