Stream ciphers

Myrto Arapinis
School of Informatics
University of Edinburgh

January 19, 2015

The One-Time Pad (OTP)

- $\mathcal{M} = \mathcal{C} = \mathcal{K} = \{0,1\}^n$
- Encryption: $\forall k \in \mathcal{K}.\ \forall m \in \mathcal{M}.\ E(k,m) = k \oplus m$

$$\begin{array}{lcllllllll} k & = & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ m & = & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ \hline c & = & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{array}$$

- Decryption: $\forall k \in \mathcal{K}.\ \forall c \in \mathcal{C}.\ D(k,c) = k \oplus c$

$$\begin{array}{lcllllllll} k & = & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ c & = & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ \hline m & = & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}$$

- Consistency: $D(k, E(k,m)) = k \oplus (k \oplus m) = m$

Perfect secrecy

Definition

A cipher (E, D) over $(\mathcal{M}, \mathcal{C}, \mathcal{K})$ satisfies perfect secrecy if for all messages $m_1, m_2 \in \mathcal{M}$ of same length ($|m_1| = |m_2|$), and for all ciphertexts $c \in \mathcal{C}$

$$|Pr(E(k, m_1) = c) - Pr(E(k, m_2) = c)| \leq \epsilon$$

where $k \xleftarrow{r} \mathcal{K}$ and ϵ is some "negligible quantity".

OTP satisfies perfect secrecy

Theorem (Shannon 1949)

The One-Time Pad satisfies perfect secrecy

Proof: We first note that for all messages $m \in \mathcal{M}$ and all ciphertexts $c \in \mathcal{C}$

$$\begin{aligned} Pr(E(k,m) = c) &= \tfrac{\#\{k \in \mathcal{K}:\ k \oplus m = c\}}{\#\mathcal{K}} \\ &= \tfrac{\#\{k \in \mathcal{K}:\ k = m \oplus c\}}{\#\mathcal{K}} \\ &= \tfrac{1}{\#\mathcal{K}} \end{aligned}$$

where $k \xleftarrow{r} \mathcal{K}$.
Thus, for all messages $m_1, m_2 \in \mathcal{M}$, and for all ciphertexts $c \in \mathcal{C}$

$$|Pr(E(k, m_1) = c) - Pr(E(k, m_2) = c)| \leq \left| \frac{1}{\#\mathcal{K}} - \frac{1}{\#\mathcal{K}} \right| = 0$$

Limitations of OTP

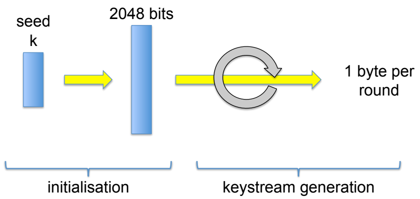
- Key-length!
  - The key should be as long as the plaintext.
- Getting true randomness!
  - The key should not be guessable from an attacker.
- Perfect secrecy does not capture all possible attacks
  - OTP is subject to two-time pad attacks
    given $m_1 \oplus k$ and $m_2 \oplus k$, we can compute
    $m_1 \oplus m_2 = (m_1 \oplus k) \oplus (m_2 \oplus k)$
    English has enough redundancy s.t. $m_1 \oplus m_2 \rightarrow m_1, m_2$
  - OTP is malleable
    given the ciphertext $c = E(k, m)$ with $m = to\ bob : m_0$, it is possible to compute the ciphertext $c' = E(k, m')$ with
    $m' = to\ eve : m_0$
    $c' := c \oplus "to\ bob : 00 \ldots 00" \oplus "to\ eve : 00 \ldots 00"$

Stream ciphers

- Goal: make the OTP practical
- Idea: use a pseudorandom key rather than a really random key
  - The key will not really be random, but will look random
  - The key will be generated from a key seed using a Pseudo-Random Generator (PRG)
    $G : \{0,1\}^s \rightarrow \{0,1\}^n$ with $s << n$
- Encryption using a PRG G: $E(k, m) = G(k) \oplus m$
- Decryption using a PRG G: $D(k, c) = G(k) \oplus c$
- Stream ciphers are subject to two-time pad attacks
- Stream ciphers are malleable

RC4

- Stream cipher invented by Ron Rivest in 1987
- Consists of 2 phases:



- Main data structure: array S of 256 bytes.
- Used in HTTPS and WEP
- Weaknesses of RC4:
  - first bytes are biased
    $\longrightarrow$ drop the first to 256 generated bytes
  - subject to related keys attacks
    $\longrightarrow$ choose randomly generated keys as seeds

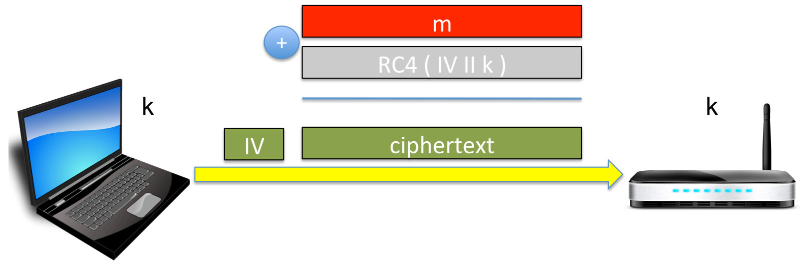
RC4: initialisation

```
for i := 0 to 255 do
    S[i] := i
end
j := 0
for i := 0 to 255 do
    j := (j + S[i] + K[i(mod |K|)])(mod 256)
    swap(S[i], S[j])
end
i := 0
j := 0
```

RC4: key stream generation

```
while generatingOutput
    i := i + 1(mod 256)
    j := j + S[i](mod 256)
    swap(S[i], S[j])
    output(S[S[i] + S[j](mod 256)])
end
```


WEP uses RC4



Initialisation Vector (IV): 24-bits long string

Weaknesses of WEP

- two-time pad attack: IV is 24 bits long, so the key is reused after at most 2^{24} frames
  $\longrightarrow$ use longer IVs
- Fluhrer, Mantin and Shamir (FMS) attack (related keys attack):
  - the keys only differ in the 24 bits IV
  - first bytes of key stream known because standard headers are always sent
  - for certain IVs knowing m bytes of key and keystream means you can deduce byte $m+1$ of key

  $\longrightarrow$ instead of using related IVs, generate IVs using a PRG

Remark

The FMS attack does not apply to RC4-based SSL (TLS), since SSL generates the encryption keys it uses for RC4 by hashing, meaning that different SSL sessions have unrelated keys

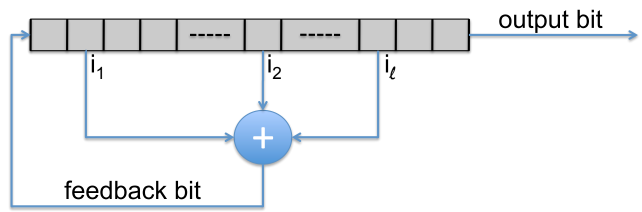
Linear Feedback Shift Registers (LFSRs)

- $\mathcal{K} = \{0,1\}^s$
- Main data structure: register R of s bits
- Initialisation: $R := k$
- Keystream generation: 1-bit output per round

  taps: $i_1, i_2, \ldots i_\ell$

  feedback bit: $R[i_1] \oplus R[i_2] \oplus \cdots \oplus R[i_\ell]$

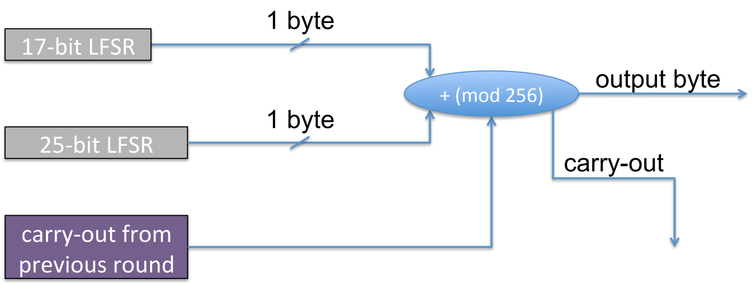
  output bit: $R[s]$



- Broken LFSR-based stream ciphers:
  - DVD encryption: CSS (2 LFSRs)
  - GSM encryption: A5 (3 LFSRs)
  - Bluetooth encryption: E0 (4 LFSRs)

Content Scrambling System (CSS) uses LFSRs

- $\mathcal{K} = \{0,1\}^{40}$
- Data structures: 17-bits LFSR (R_{17}) and 25-bits LFSR (R_{25})
- Initialisation: $R_{17} := 1||K[0-15]$
  $R_{25} := 1||K[16-39]$
- Keystream generation: 1-byte output per round



Weaknesses in CSS

Can be broken in time 2^{17}. The idea of the attack is as follows:

- Because of structure of MPEG-2, first 20 bytes of plaintext are known
- Hence also first 20 bytes of keystream are known
- Given output of 17 bit LFSR, can deduce output of 25 bit LFSR by subtraction
- Hence try all 2^{17} possibilities for 17 bit LFSR and if generated 25 bit LFSR produces observed keystream, cipher is cracked

Modern stream ciphers

Project eStream: project to "identify new stream ciphers suitable for widespread adoption", organised by the EU ECRYPT network
$\longrightarrow$ HC-128, Rabbit, Salsa20/12, SOSEMANUK,
Grain v1, MICKEY 2.0, Trivium

Conjecture

These eStream stream ciphers are "secure"

Concluding remarks

- Perfect secrecy does not capture all possible attacks.
  $\longrightarrow$ need for different security definition
- Theorem (Shannon 1949) Let (E, D) be a cipher over $(\mathcal{M}, \mathcal{C}, \mathcal{K})$. If (E, D) satisfies perfect secrecy, then the keys should be at least as long as the plaintexts ($|\mathcal{K}| \leq |\mathcal{M}|$).
  $\Rightarrow$ Stream ciphers do not satisfy perfect secrecy because the keys in $\mathcal{K}$ are smaller than the messages in $\mathcal{M}$
  $\longrightarrow$ need for different security definition
- The design of crypto primitives is a subtle and error prone task: define threat model, propose construction, prove that breaking construction would solve an underlying hard problem.
  $\longrightarrow$ use standardised publicly know primitives
- Crypto primitives are secure under a precisely defined threat model.
  $\longrightarrow$ respect the security assumptions of the crypto primitives you use