

# Angriffe auf RC4

Andreas Klein

Universität Kassel

Tagung der Fachgruppe Computeralgebra, Kassel 2005

# Überblick

- 1 Der RC4 Algorithmus
  - Geschichte
  - Beschreibung des Algorithmus
  - Angriffe

# Überblick

- 1 Der RC4 Algorithmus
  - Geschichte
  - Beschreibung des Algorithmus
  - Angriffe
- 2 Korrelationen in RC4
  - Überblick
  - Beispiel

# Überblick

- 1 Der RC4 Algorithmus
  - Geschichte
  - Beschreibung des Algorithmus
  - Angriffe
- 2 Korrelationen in RC4
  - Überblick
  - Beispiel
- 3 Angriffe auf RC4
  - Entwicklung eines Angriffs
  - Vergleich mit der FMS-Attacke

# Überblick

- 1 Der RC4 Algorithmus
  - Geschichte
  - Beschreibung des Algorithmus
  - Angriffe
- 2 Korrelationen in RC4
  - Überblick
  - Beispiel
- 3 Angriffe auf RC4
  - Entwicklung eines Angriffs
  - Vergleich mit der FMS-Attacke
- 4 Fazit

# Geschichte des RC4

## Entwicklung

- Entwickelt 1987 von Ron Rivest
- Firmengeheimnis von RSA Data Security

# Geschichte des RC4

## Entwicklung

- Entwickelt 1987 von Ron Rivest
- Firmengeheimnis von RSA Data Security
- 1994 anonyme Veröffentlichung auf der Mailing-Liste Cypherpunks

# Geschichte des RC4

## Entwicklung

- Entwickelt 1987 von Ron Rivest
- Firmengeheimnis von RSA Data Security
- 1994 anonyme Veröffentlichung auf der Mailing-Liste Cypherpunks

## Einsatzgebiete

# Geschichte des RC4

## Entwicklung

- Entwickelt 1987 von Ron Rivest
- Firmengeheimnis von RSA Data Security
- 1994 anonyme Veröffentlichung auf der Mailing-Liste Cypherpunks

## Einsatzgebiete

- WEP (WLAN Verschlüsselung)

# Geschichte des RC4

## Entwicklung

- Entwickelt 1987 von Ron Rivest
- Firmengeheimnis von RSA Data Security
- 1994 anonyme Veröffentlichung auf der Mailing-Liste Cypherpunks

## Einsatzgebiete

- WEP (WLAN Verschlüsselung)
- Oracle Secure SQL

# Geschichte des RC4

## Entwicklung

- Entwickelt 1987 von Ron Rivest
- Firmengeheimnis von RSA Data Security
- 1994 anonyme Veröffentlichung auf der Mailing-Liste Cypherpunks

## Einsatzgebiete

- WEP (WLAN Verschlüsselung)
- Oracle Secure SQL
- Apple AOCE

# Der RC4 Algorithmus

## Struktur

- RC4 erzeugt eine Folge von Pseudozufalls-Bytes.
- Geheime Permutation  $S$  der Zahlen von 0 bis 255
- Geheimen Zeiger  $j$ , öffentlichen Zeiger  $i$

# Der RC4 Algorithmus

## Struktur

- RC4 erzeugt eine Folge von Pseudozufalls-Bytes.
- Geheime Permutation  $S$  der Zahlen von 0 bis 255
- Geheimen Zeiger  $j$ , öffentlichen Zeiger  $i$

## Verschlüsselung

```
i = (i + 1) mod 256  
j = (j + S[i]) mod 256  
vertausche S[i] und S[j]  
k = (S[i]+S[j]) mod 256  
Ausgabe S[k]
```

# Die Schlüsselerzeugung

## Struktur

- Typischer Weise setzt sich der Sitzungsschlüssel aus Initialisierungsvektor und Hauptschlüssel zusammen.
- Der Sitzungsschlüssel wird durch periodisches Wiederholen auf die Länge 255 erweitert.
- Die Permutation  $S$  wird mit der Identität initialisiert.

# Die Schlüsselerzeugung

## Struktur

- Typischer Weise setzt sich der Sitzungsschlüssel aus Initialisierungsvektor und Hauptschlüssel zusammen.
- Der Sitzungsschlüssel wird durch periodisches Wiederholen auf die Länge 255 erweitert.
- Die Permutation  $S$  wird mit der Identität initialisiert.

## Erzeugung der Anfangspermutation

```
j = 0;  
FOR i FROM 0 TO 255:  
  j = (j + S[i] + K[i]) mod 256  
  vertausche S[i] und S[j]
```

# Angriffe auf RC4

## Auswahl von Veröffentlichungen

- Jovan Dj. Golić, 1997, Eine Folge von  $2^{40}$  Pseudozufallsbytes kann von echten Zufallszahlen unterschieden werden.
- Scott R. Fluhrer und David A. McGrew, 2000: Starke Korrelationen in der Ausgabe
- Ilya Mironov, 2002: Schwächen in der Schlüsselerzeugung
- S. Fluhrer, I. Martin, A. Shamir, 2001: stärkster bislang bekannter Angriff, gewählte Initialisierungsvektoren.

# Überblick

## Schwächen im RC4 Schlüsselstrom

# Überblick

## Schwächen im RC4 Schlüsselstrom

- (Fluhrer, McGrew): Bigramm-Wahrscheinlichkeiten

# Überblick

## Schwächen im RC4 Schlüsselstrom

- (Fluhrer, McGrew): Bigramm-Wahrscheinlichkeiten
- Ist die Ausgabe zum Zeitpunkt  $t$  gleich  $t - 1$ , so ist die Ausgabe zum Zeitpunkt  $t + 256$  gleich 1 mit Wahrscheinlichkeit  $\frac{1}{256} + \frac{1}{256^2}$ !

# Überblick

## Schwächen im RC4 Schlüsselstrom

- (Fluhrer, McGrew): Bigramm-Wahrscheinlichkeiten
- Ist die Ausgabe zum Zeitpunkt  $t$  gleich  $t - 1$ , so ist die Ausgabe zum Zeitpunkt  $t + 256$  gleich 1 mit Wahrscheinlichkeit  $\frac{1}{256} + \frac{1}{256^2}$ !
- Es gilt

$$P(S[j] + S[k] \equiv i \pmod{256}) = \frac{2}{256}$$

## Beispiel

### Satz

*Unter der Annahme, daß alle internen Zustände gleichwahrscheinlich sind, gilt*

$$P(S[j] + S[k] \equiv i \pmod{256}) = \frac{2}{256}$$

## Beispiel

### Satz

*Unter der Annahme, daß alle internen Zustände gleichwahrscheinlich sind, gilt*

$$P(S[j] + S[k] \equiv i \pmod{256}) = \frac{2}{256}$$

### Beweisidee

Zähle alle Permutationen, bei denen  $S[j] + S[k] \equiv i \pmod{256}$  gilt. Führe eine Fallunterscheidung durch, ob  $i$ ,  $j$  oder  $k$  zusammenfallen.

# Fall $i = j$

## Fall $i = j$

①  $2S[i] = i$

Gilt  $j = i$  und  $2S[i] = i$ , so ist  $k = i$  und damit  $S[k] = S[i]$ , d.h.  $S[k] + S[i] = i$  wie gewünscht. Es gibt in diesem Fall noch  $255!$  Möglichkeiten die Permutation zu ergänzen.

## Fall $i = j$

①  $2S[i] = i$

Gilt  $j = i$  und  $2S[i] = i$ , so ist  $k = i$  und damit  $S[k] = S[i]$ , d.h.  $S[k] + S[i] = i$  wie gewünscht. Es gibt in diesem Fall noch  $255!$  Möglichkeiten die Permutation zu ergänzen.

②  $2S[i] \neq i$

In diesem Fall muß mit  $k = S[i]$  die Gleichung  $S[k] = i - S[i] \pmod n$  gelten. Damit sind  $S[i]$  und  $S[k]$  fest gelegt und es gibt noch  $254!$  Möglichkeiten die Permutation zu ergänzen.

# Fall $i \neq j$

## Fall $i \neq j$

①  $2S[j] = i$

In diesem Fall muß  $k = S[i] + S[j] = j$  gelten. Damit ist  $S[i]$  festgelegt und es gibt  $254!$  Möglichkeiten die Permutation zu ergänzen.

## Fall $i \neq j$

①  $2S[j] = i$

In diesem Fall muß  $k = S[i] + S[j] = j$  gelten. Damit ist  $S[i]$  festgelegt und es gibt  $254!$  Möglichkeiten die Permutation zu ergänzen.

②  $2S[j] \neq i$

## Fall $i \neq j$

①  $2S[j] = i$

In diesem Fall muß  $k = S[i] + S[j] = j$  gelten. Damit ist  $S[i]$  festgelegt und es gibt  $254!$  Möglichkeiten die Permutation zu ergänzen.

②  $2S[j] \neq i$

①  $S[i] + S[j] = i$

Es gibt in diesem Fall 255 verschiedene Möglichkeiten  $j$  zu wählen. Damit sind  $S[i]$  und  $S[j]$  festgelegt und es bleiben noch  $254!$  Möglichkeiten die restliche Permutation zu wählen.

## Fall $i \neq j$

①  $2S[j] = i$

In diesem Fall muß  $k = S[i] + S[j] = j$  gelten. Damit ist  $S[i]$  festgelegt und es gibt  $254!$  Möglichkeiten die Permutation zu ergänzen.

②  $2S[j] \neq i$

①  $S[i] + S[j] = i$

Es gibt in diesem Fall 255 verschiedene Möglichkeiten  $j$  zu wählen. Damit sind  $S[i]$  und  $S[j]$  festgelegt und es bleiben noch  $254!$  Möglichkeiten die restliche Permutation zu wählen.

②  $S[i] + S[j] \neq i$

Dann muß  $S[i] + S[j] \neq j$  gelten. Hat man  $S[j]$  gewählt ist auch  $S[k]$  mit  $k = S[i] + S[j]$  vorgeben und es gibt nur  $253!$  Möglichkeiten die Permutation zu ergänzen.  
 $254 \cdot 253 \cdot 253! + 254!$  Möglichkeiten.

# Ziel und Voraussetzungen

Ziel des Angriffs

## Ziel und Voraussetzungen

### Ziel des Angriffs

- Der Angreifer erfährt die Summe der ersten beiden Bytes des Hauptschlüssels.

### Voraussetzungen des Angriffs

## Ziel und Voraussetzungen

### Ziel des Angriffs

- Der Angreifer erfährt die Summe der ersten beiden Bytes des Hauptschlüssels.

### Voraussetzungen des Angriffs

- Die Sitzungsschlüssel haben die Form  
Hauptschlüssel||Initialisierungsvektor.

# Ziel und Voraussetzungen

## Ziel des Angriffs

- Der Angreifer erfährt die Summe der ersten beiden Bytes des Hauptschlüssels.

## Voraussetzungen des Angriffs

- Die Sitzungsschlüssel haben die Form  
Hauptschlüssel||Initialisierungsvektor.
- Der Initialisierungsvektor ist dem Angreifer bekannt.

# Ziel und Voraussetzungen

## Ziel des Angriffs

- Der Angreifer erfährt die Summe der ersten beiden Bytes des Hauptschlüssels.

## Voraussetzungen des Angriffs

- Die Sitzungsschlüssel haben die Form  
Hauptschlüssel||Initialisierungsvektor.
- Der Initialisierungsvektor ist dem Angreifer bekannt.
- Der Angreifer kann 14000 Verschlüsselungen abhören.

## Ziel und Voraussetzungen

### Ziel des Angriffs

- Der Angreifer erfährt die Summe der ersten beiden Bytes des Hauptschlüssels.

### Voraussetzungen des Angriffs

- Die Sitzungsschlüssel haben die Form Hauptschlüssel||Initialisierungsvektor.
- Der Initialisierungsvektor ist dem Angreifer bekannt.
- Der Angreifer kann 14000 Verschlüsselungen abhören.
- Das jeweils erste Byte des Schlüsselstroms ist dem Angreifer bekannt.

# Durchführung des Angriffs

## Durchführung des Angriffs

- In den ersten zwei Schritten der Schlüsselerzeugung wird  $S[1]$  auf  $t = K[0] + K[1] + 1$  gesetzt. (Gilt nur falls  $K[0] \neq 1$ .)

## Durchführung des Angriffs

- In den ersten zwei Schritten der Schlüsselerzeugung wird  $S[1]$  auf  $t = K[0] + K[1] + 1$  gesetzt. (Gilt nur falls  $K[0] \neq 1$ .)
- Mit der Wahrscheinlichkeit  $(1 - \frac{1}{256})^{254} \approx \frac{1}{e}$  wird dieser Wert nicht mehr verändert.

## Durchführung des Angriffs

- In den ersten zwei Schritten der Schlüsselerzeugung wird  $S[1]$  auf  $t = K[0] + K[1] + 1$  gesetzt. (Gilt nur falls  $K[0] \neq 1$ .)
- Mit der Wahrscheinlichkeit  $(1 - \frac{1}{256})^{254} \approx \frac{1}{e}$  wird dieser Wert nicht mehr verändert.
- Im ersten Schritt ( $i = 1$ ) der Verschlüsselung ergibt sich vermutlich durch das Vertauschen von  $S[1]$  und  $S[j]$  die Gleichung  $S[j] = t$ .

## Durchführung des Angriffs

- In den ersten zwei Schritten der Schlüsselerzeugung wird  $S[1]$  auf  $t = K[0] + K[1] + 1$  gesetzt. (Gilt nur falls  $K[0] \neq 1$ .)
- Mit der Wahrscheinlichkeit  $(1 - \frac{1}{256})^{254} \approx \frac{1}{e}$  wird dieser Wert nicht mehr verändert.
- Im ersten Schritt ( $i = 1$ ) der Verschlüsselung ergibt sich vermutlich durch das Vertauschen von  $S[1]$  und  $S[j]$  die Gleichung  $S[j] = t$ .
- Es gilt mit hoher Wahrscheinlichkeit  $S[j] \equiv 1 - S[K] \pmod{256}$ .

## Durchführung des Angriffs

- In den ersten zwei Schritten der Schlüsselerzeugung wird  $S[1]$  auf  $t = K[0] + K[1] + 1$  gesetzt. (Gilt nur falls  $K[0] \neq 1$ .)
- Mit der Wahrscheinlichkeit  $(1 - \frac{1}{256})^{254} \approx \frac{1}{e}$  wird dieser Wert nicht mehr verändert.
- Im ersten Schritt ( $i = 1$ ) der Verschlüsselung ergibt sich vermutlich durch das Vertauschen von  $S[1]$  und  $S[j]$  die Gleichung  $S[j] = t$ .
- Es gilt mit hoher Wahrscheinlichkeit  $S[j] \equiv 1 - S[K] \pmod{256}$ .
- Dies liefert eine Schätzung für  $t$ .

## Vergleich mit der FMS-Attacke

Fluhrer, Martin, Shamir (FMS) Angriff

neue Angriff

## Vergleich mit der FMS-Attacke

### Fluhrer, Martin, Shamir (FMS) Angriff

- Man kann ein beliebiges Bytes des Hauptschlüssels ermitteln.

### neue Angriff

- Nur die Summe der ersten zwei Bytes wird ermittelt.

## Vergleich mit der FMS-Attacke

### Fluhrer, Martin, Shamir (FMS) Angriff

- Man kann ein beliebiges Bytes des Hauptschlüssels ermitteln.
- Sitzungsschlüssel = Initialisierungsvektor||Hauptschlüssel.

### neue Angriff

- Nur die Summe der ersten zwei Bytes wird ermittelt.
- Sitzungsschlüssel = Hauptschlüssel||Initialisierungsvektor.

## Vergleich mit der FMS-Attacke

### Fluhrer, Martin, Shamir (FMS) Angriff

- Man kann ein beliebiges Bytes des Hauptschlüssels ermitteln.
- Sitzungsschlüssel = Initialisierungsvektor||Hauptschlüssel.
- Zwei Bytes des Initialisierungsvektors werden vorgegeben.

### neue Angriff

- Nur die Summe der ersten zwei Bytes wird ermittelt.
- Sitzungsschlüssel = Hauptschlüssel||Initialisierungsvektor.
- Keine Anforderungen an den Initialisierungsvektor.

## Vergleich mit der FMS-Attacke

### Fluhrer, Martin, Shamir (FMS) Angriff

- Man kann ein beliebiges Bytes des Hauptschlüssels ermitteln.
- Sitzungsschlüssel = Initialisierungsvektor||Hauptschlüssel.
- Zwei Bytes des Initialisierungsvektors werden vorgegeben.
- Benötigte Anzahl von Verschlüsselungen  $\approx 60$ .

### neue Angriff

- Nur die Summe der ersten zwei Bytes wird ermittelt.
- Sitzungsschlüssel = Hauptschlüssel||Initialisierungsvektor.
- Keine Anforderungen an den Initialisierungsvektor.
- Benötigte Anzahl von Verschlüsselungen  $\approx 14000$ .

# Fazit

# Fazit

- 1 Die von RC4 erzeugte Pseudozufallsfolge ist relativ schwach.

# Fazit

- 1 Die von RC4 erzeugte Pseudozufallsfolge ist relativ schwach.
- 2 Zusammen mit Schwächen der Schlüsselerzeugung, entstehen praktikable Angriffe.

## Fazit

- 1 Die von RC4 erzeugte Pseudozufallsfolge ist relativ schwach.
- 2 Zusammen mit Schwächen der Schlüsselerzeugung, entstehen praktikable Angriffe.
- 3 Verbesserungsvorschlag (Mironov 2002): Die ersten  $12 \cdot 256$  Bytes des RC4 Schlüsselstroms sollten nicht benutzt werden.

## Fazit

- 1 Die von RC4 erzeugte Pseudozufallsfolge ist relativ schwach.
- 2 Zusammen mit Schwächen der Schlüsselerzeugung, entstehen praktikable Angriffe.
- 3 Verbesserungsvorschlag (Mironov 2002): Die ersten  $12 \cdot 256$  Bytes des RC4 Schlüsselstroms sollten nicht benutzt werden. Achtung: Das Weglassen von nur 256 Bytes reicht nicht!